



## **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

“ANÁLISIS Y PROPUESTA DE UN SISTEMA  
CONTROLADOR DE CÓDIGO ABIERTO PARA LA RED  
INALÁMBRICA EN LA FACULTAD DE INGENIERÍA EN  
ELECTRICIDAD Y COMPUTACIÓN DE LA ESPOL”

### **INFORME DE PROYECTO INTEGRADOR**

Previo a la obtención del Título de:

**INGENIERÍA EN TELEMÁTICA**

ISAAC JOSUE BALLÉN GAVILÁNEZ

CHRISTIAN IGNACIO BAIDAL VELOZ

GUAYAQUIL – ECUADOR

AÑO: 2018

## **AGRADECIMIENTOS**

Primero agradezco a Dios por haberme guiado en el transcurso de este objetivo anhelado y brindarme su fortaleza en los momentos de apremio y debilidad, segundo a mi madre, Victoria Gavilánez, por apoyarme en todo momento y por su esfuerzo diario por darme una excelente educación e inculcarme buenos valores que han forjado la persona que soy en la actualidad, también quiero agradecer a una persona muy especial en mi vida, mi novia Ivette Villón por su gran apoyo en este largo camino a culminar, que con su compañía, paciencia, amor y buenos consejos me han ayudado a superar varios obstáculos en los momentos más difíciles.

Agradezco igualmente a los tutores de este proyecto integrador Ing. Gabriel Astudillo e Ing. Benjamin Flament por brindarme la oportunidad de trabajar con ustedes, ofreciendo sus conocimientos para el desarrollo de este proyecto.

Isaac Ballén

## **AGRADECIMIENTOS**

En primer lugar, quiero empezar agradeciendo a Dios por guiar mis pasos y darme salud en cada instante de mi vida, a mi madre por inculcarme valores, darme consejos siendo un gran ejemplo en mi vida, a mi padre por estar en todo momento brindándome su apoyo en mis estudios y en la vida, acompañándome en todas mis necesidades y problemas, a mi hermana por estar siempre ahí en las buenas y en las malas conmigo y a mi hermano por darme la mano cuando más la necesite.

Christian Baidal

## **DEDICATORIA**

El esfuerzo, la paciencia y la dedicación a lo largo de este anhelado objetivo se lo dedico a Dios, por haberme permitido llegar a este punto en mi vida, asimismo de su gran bondad e infinito amor. A mi madre Victoria Gavilanes por ser un pilar fundamental en mi vida y por su apoyo incondicional en los malos y buenos momentos. A mi novia Ivette Villón por su gran motivación, confianza y ayuda fundamental en tiempos de mayor dificultad brindándome su amor y comprensión e impulsándome a ser mejor cada día. A nuestros tutores por su apoyo y tiempo compartido en el desarrollo de este proyecto.

Isaac Ballén

## **DEDICATORIA**

Todo el esfuerzo y dedicatoria está dirigido a Dios y el amor incondicional de mi madre Bella Veloz, pilar fundamental en todo este transcurso de mi carrera como estudiante ya que ella se sacrificó día a día para darme lo mejor y no me falte nada, a mi padre Johnny Baidal y mis hermanos que me dieron la confianza y apoyo en cada momento en que los necesite, y a mis maestros por hacer posible que este proyecto se realice con éxito.

Christian Baidal

# TRIBUNAL DE EVALUACIÓN

.....  
**Ing. Gabriel Astudillo, M.Sc.**

PROFESOR COLABORADOR

.....  
**Ing. Benjamin Flament, M.Sc.**

PROFESOR TUTOR

## **DECLARACIÓN EXPRESA**

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

.....  
**Isaac Josué Ballén Gavilanes**

.....  
**Christian Ignacio Baidal Veloz**

## RESUMEN

Este trabajo describe una solución en software de código abierto compatible con diferentes modelos de APs para su administración dentro de una red inalámbrica en la Facultad de Ingeniería en Electricidad y Computación (FIEC).

En este proyecto se utilizaron varias herramientas tales como el sistema operativo Ubuntu server que contiene al software controlador, Routers inalámbricos de diferentes fabricantes TP-LINK Y CISCO para realizar las pruebas respectivas, Openwrt que es el firmware base de los Routers inalámbricos, Ansible que permite la configuración de servidores de manera remota, Playbook para la configuración de la controladora y el paquete openwisp-config-openssl que ayudará al registro de los dispositivos al software controlador.

El inicio del proyecto consiste en la instalación del sistema operativo más adecuado para el software controlador, hecho esto se procede a la instalación y configuración de Openwisp2 Controller que será la controladora que administre los Routers inalámbricos, para esto se necesita instalar el firmware de Openwrt en los Routers verificando la marca y versión de estos en la página oficial [1], cabe recalcar que la mayoría de los Routers son compatibles con versiones de Openwrt, instalado el firmware se requiere descargar un paquete adicional llamado openwisp-config-openssl el cual se necesita configurar mediante línea de comandos entrando al dispositivo vía remota, esta paquetería permitirá que los dispositivos se registren en Openwisp2 Controller y tome el control estos mediante la creación de plantillas dependiendo el de lo que se requiera como por ejemplo modificar sus interfaces de red o configurar una red inalámbrica, por último se agregaran planos de la ubicación de los dispositivos para saber su localización dentro de una área geográfica.

Palabras Claves: Controlador, Código abierto, APs, Openwrt.

## ABSTRACT

*This paper describes a solution in open source software compatible with different APs models for management within a wireless network at the Faculty of Electrical and Computer Engineering (FIEC).*

*In this project were used several tools such as the operating system Ubuntu server that contains the software controller, wireless routers of different manufacturers TP-LINK and CISCO to perform the respective tests, Openwrt which is the firmware Base of wireless routers, ansible that allows the configuration of servers remotely, Playbook for the configuration of the controller and the package Openwisp-config-OpenSSL that will help to register devices to the driver software.*

*The start of the project consists in the installation of the most suitable operating system for the Controller software, this is done by the installation and configuration of OPENWISP2 controller that will be the controller that manages the wireless routers, for this You need to install the firmware of OpenWrt on the routers checking the brand and version of these on the official page [1], it is important to emphasize that most of the routers are compatible with versions of OpenWrt, installed the firmware requires downloading an additional package called Openwisp-Config-openssl which needs to be configured via command line entering the device via remote, this package will allow devices to register in OPENWSIP2 Controller and take control these by creating templates depending on what is required such as modifying your network interfaces or configuring a wireless network, finally you will add blueprints to the location of the devices to know your location within a geographic area.*

*Keywords: Controller, Open Source, APs, Openwrt.*

## ABREVIATURA

SSID	Service Set Identifier, nombre que identifica una red inalámbrica.
HTTPS	Protocolo seguro de transferencia de hipertexto.
SSL	Secure Sockets Layer.
AP	Punto de acceso Inalámbrico.
SSH	Secure Shell, protocolo de administración remota.
URL	Uniform Resource Locator.
WIFI	Tecnología que permite la interconexión de dispositivos inalámbricos.
IP	Protocolo de internet
OPENWRT	Firmware basado en Linux para dispositivos embebidos.
PUTTY	Emulador de terminal que soporte conexiones mediante ssh, telnet.

## ÍNDICE DE FIGURAS

Figura 2.1 OpenWISP2 .....	4
Figura 2.2 Cable Ethernet .....	6
Figura 2.3 Router TL – WR1043ND. ....	7
Figura 2.4 Cisco Cisco E4200. ....	8
Figura 2.5 Switch TL – SF1008D. ....	9
Figura 2.6 Adaptador USB a Ethernet. ....	9
Figura 3.1 Configuración de las interfaces de redes.....	11
Figura 3.2 Creación de usuarios .....	11
Figura 3.3 Crear las particiones del disco. ....	12
Figura 3.4 Interfaz de lista de comandos.....	12
Figura 3.5 Controladores Open Source.....	13
Figura 3.6 Instalación de Ansible. ....	14
Figura 3.7 Instalación de Git .....	14
Figura 3.8 Instalación de openwisp.openwisp2. ....	15
Figura 3.9 Creación del directorio de trabajo.....	15
Figura 3.10 Creación del archivo hosts. ....	16
Figura 3.11 Creación del archivo playbook.yml. ....	17
Figura 3.12 Ejecución de los archivos hosts y playbook.yml. ....	17
Figura 3.13 Instalación y configuración de Openwisp2 Controller. ....	18
Figura 3.14 Inicio de Sesión de Openwisp2 Controller. ....	18
Figura 3.15 Openwisp2 Controller.....	19
Figura 3.16 Página oficial de Openwrt. ....	20
Figura 3.17 Firmware de Openwrt para el Router WR1043ND.....	20
Figura 3.18 Símbolo de sistema.....	20
Figura 3.19 Panel de administración de Router WR1043ND.....	21
Figura 3.20 Instalación del Firmware Openwrt. ....	21
Figura 3.21 Símbolo de sistema Openwrt. ....	22
Figura 3.22 Interfaz de Openwrt router WR1043ND.....	22
Figura 3.23 Putty Configuration.....	23
Figura 3.24 Ventana de ingreso al router por ssh.....	23
Figura 3.25 Página oficial de Openwisp. ....	24
Figura 3.26 downloads.openwisp.org.....	24

Figura 3.27 openwisp-config-openssl.....	25
Figura 3.28 Archivo openwisp.....	26
Figura 3.29 Configuración de option url.....	26
Figura 3.30 Creación de una Organización.....	27
Figura 3.31 Creación de Shared secret.....	27
Figura 3.32 Campo option shared_secret.....	27
Figura 3.33 Shared secret por defecto.....	28
Figura 3.34 Archivo openwisp configurado.....	28
Figura 4.1 Diagrama de red Openwisp2 Controller.....	29
Figura 4.2 Registro del dispositivo a Openwisp2 Controller.....	30
Figura 4.3 Pestaña Devices de Openwisp2 Controller.....	30
Figura 4.4 Dispositivo TP-LINK TL-WR1043ND registrado.....	31
Figura 4.5 Información general del router TP-LINK TL-WR1043ND.....	31
Figura 4.6 Cambio de nombre al router TP-LINK TL-WR1043ND.....	31
Figura 4.7 Registro del dispositivo Cisco a Openwisp2 Controller.....	32
Figura 4.8 Dispositivo Cisco Linksys E4200 registrado.....	32
Figura 4.9 Agregando una plantilla.....	33
Figura 4.10 Configuración del nombre a la Plantilla.....	33
Figura 4.11 Configuración de la Zona horaria.....	34
Figura 4.12 Menú de plantillas.....	34
Figura 4.13 Configuration Menú.....	35
Figura 4.14 Creación de una interfaz de red.....	35
Figura 4.15 Configuración de una interfaz de red.....	36
Figura 4.16 Configuración de la seguridad de la red.....	37
Figura 4.17 Configuración de Radios plantilla WIFI-TPLINK.....	38
Figura 4.18 Uso de la plantilla WIFI-TPLINK.....	39
Figura 4.19 Logs del dispositivo TP-LINK TL-WR1043ND.....	39
Figura 4.20 Wireless Overview TP-LINK TL-WR1043ND.....	40
Figura 4.21 Device Configuration TP-LINK TL-WR1043ND.....	40
Figura 4.22 Configuración de Radios plantilla WIFI-CISCO.....	41
Figura 4.23 Uso de la plantilla WIFI-CISCO.....	41
Figura 4.24 Logs del dispositivo CISCO-LINKSYS-E4200.....	42
Figura 4.25 Wireless Overview CISCO-LINKSYS-E4200.....	42
Figura 4.26 Red inalámbrica FIEC.....	43
Figura 4.27 Ingreso a la Red FIEC.....	43

Figura 4.28 Conexión a la Red FIEC.....	44
Figura 4.29 Red FIEC router TP-LINK.....	44
Figura 4.30 Red FIEC router CISCO.....	45
Figura 4.31 GEOGRAPHIC ADMINISTRATION.....	45
Figura 4.32 Creación de mapa de los routers.....	46
Figura 4.33 Creación de un Floor plans. ....	46
Figura 4.34 Location selection. ....	47
Figura 4.35 Indoor Coordinates.....	47

# ÍNDICE GENERAL

RESUMEN.....	i
ABSTRACT.....	ii
ABREVIATURA.....	iii
ÍNDICE DE FIGURAS.....	iv
ÍNDICE GENERAL.....	1
CAPÍTULO 1.....	2
1. PLANTEAMIENTO DEL PROBLEMA.....	2
1.3.1 Objetivo general.....	3
1.3.2 Objetivos específicos.....	3
CAPÍTULO 2.....	4
2. MARCO TEÓRICO.....	4
2.1.1 OpenWISP2 Controller.....	4
2.1.2 OpenWISP2 Firmware.....	4
2.1.3 OpenWISP2 Monitoring.....	5
CAPÍTULO 3.....	10
3. IMPLEMENTACIÓN Y DESARROLLO.....	10
3.1.1 Instalación de la interfaz gráfica de Ubuntu server.....	12
3.2.1 Instalación de Ansible.....	14
3.2.2 Instalando el rol Openwisp2.....	15
3.2.3 Directorio de Trabajo.....	15
3.2.4 Archivo de Inventario.....	16
3.2.5 Creación del archivo Playbook.....	16
3.2.6 Ejecutando el archivo hosts y Playbook.yml.....	17
3.5.1 Configuración de openwisp-config-openssl.....	25
CAPÍTULO 4.....	29
4. PRUEBAS Y RESULTADOS.....	29
4.2.1 Plantilla Zona Horaria.....	33
4.2.2 Plantilla WIFI.....	34
4.2.3 Pruebas de la red inalámbrica FIEC.....	42
CONCLUSIONES Y RECOMENDACIONES.....	48
BIBLIOGRAFÍA.....	50

# CAPÍTULO 1

## 1. PLANTEAMIENTO DEL PROBLEMA.

Es importante destacar que la Facultad de Ingeniería en Electricidad y Computación (FIEC) cuenta con varios puntos de acceso (APs) dentro y fuera de los edificios para brindar el servicio de internet al personal docente, administrativo y estudiantes.

A pesar de todo en los últimos años el servicio que se brinda es inestable y causa inconformidad en los usuarios (profesores, estudiantes) debido a la frecuentes desconexiones que sufren al momento de trasladarse de un lugar a otro dentro de la facultad ya que, al perder señal con la red inalámbrica los usuarios tienen que volver autenticarse y el cliente se encuentra con dos redes distintas con diferentes SSID y métodos de autenticación lo cual es molesto y causa confusión ya que el usuario no sabe a qué AP conectarse. Claro está que en la zona existen otros APs que no pertenecen a la facultad, así como también una zona geográfica cubierta de árboles y edificios los cuales causan interferencia en la señal y por esta razón la conexión puede ser interrumpida.

En años anteriores se realizaron análisis e investigaciones para resolver la presente problemática, cuyas soluciones apuntaban a la incorporación de una controladora de APs para la red de la facultad, pero por su alto costo, estaba fuera del presupuesto de la universidad por lo cual no se llevaban en marcha dichas soluciones, de esta manera es preciso realizar la búsqueda de un software de bajo costo o de código abierto para ayudar a disminuir costos. Por tal motivo resulta relevante analizar esta problemática para que en un futuro no solo se pueda implementar en la FIEC sino en toda la universidad.

### 1.1 Descripción del problema

La FIEC no cuenta con un sistema controlador de APs para dispositivos de distintos fabricantes en su red inalámbrica, debido al alto costo en el mercado, generando la existencia de varios SSID y carencia de uniformidad, ocasionando disconformidad y malestar de profesores y estudiantes al acceder al servicio de internet.

Se necesita de un software controlador que administre los APs de FIEC para que trabajen de manera unificada, de tal manera el administrador de red obtendrá la flexibilidad de configurarlos desde un software principal sin necesidad de realizarlo manualmente uno a uno, además los usuarios tendrán la ventaja de ver un solo SSID en toda la facultad.

Para la propuesta de este sistema se debe tomar en cuenta los siguientes puntos:

- Debe ser compatible con APs de diferentes fabricantes en el mercado, ya que la facultad dispone de varias marcas y modelos de equipos.
- El sistema debe trabajar con el método de comunicación Maestro-Esclavo, en el primer caso sería la controladora y en el segundo los APs.
- El software debe soportar diferentes protocolos a nivel de capas como 802.11a, 802.11b, 802.11g, 802.11n entre los más importantes adicionalmente a protocolos de seguridad como SSH, SSL, HTTPS.
- Interfaz gráfica agradable y entendible por el usuario.

## **1.2 Justificación del problema**

El presente proyecto se enfoca en la búsqueda e implementación de una controladora de código abierto para APs la cual va permitir dar una solución más factible, económica y funcionalmente a la problemática que existe y así administrar la red inalámbrica de la FIEC, la cual a largo plazo podría ser implementada en el resto de facultades de la universidad brindando un buen funcionamiento a un costo relativamente bajo.

## **1.3 Objetivos**

### **1.3.1 Objetivo general**

Analizar y proponer un sistema controlador para la red inalámbrica de la Facultad de Ingeniería en Electricidad y Computación que ofrezca un correcto funcionamiento en el control y configuración de los APs de la facultad.

### **1.3.2 Objetivos específicos.**

- Analizar las distintas herramientas de código abierto que ofrezcan funcionar como un controlador de red inalámbrico.

- Lograr configurar un único SSID a todos los AP para no generar confusión al momento de elegir a que red se conecta el usuario.
- Instalar el sistema operativo más adecuado para instalar la controladora con su entorno web.
- Búsqueda e instalación del firmware compatible con los Routers de diferentes fabricantes.
- Instalación de los paquetes de comunicación con la controladora en cada uno de los Routers.

#### **1.4 Limitaciones y alcance del sistema**

- La solución de la problemática abarca solamente la FIEC por el momento.
- Debido a que la facultad no cuenta con recursos económicos se va a trabajar con software de código abierto para la controladora.
- No todos los Routers de las distintas marcas y modelos cuentan con un firmware compatible para comunicarse con la controladora.
- Al ser una herramienta de código abierto, en caso de experimentar un problema o fallo, no se contará con un soporte técnico que lo solucione, sino que se tendrá que obligadamente buscar la solución en la comunidad de dicho software.

## CAPÍTULO 2

### 2. MARCO TEÓRICO

#### 2.1 OpenWISP2

Es un software contenido por un grupo de módulos creados para agilizar y automatizar la administración de redes, orientado especialmente en redes inalámbricas, y ante todo utilizado en wifi público, redes en malla, redes comunitarias y escenarios de IoT, siendo su principal objetivo el implementar un servicio completo de Wi-Fi. [2].



Figura 2.1 OpenWISP2

##### 2.1.1 OpenWISP2 Controller

Es un Administrador de configuración y controlador de dispositivos con OpenWRT y LEDE, realizando una gestión centralizada de grandes cantidades de APs, todos con una versión de firmware OpenWRT (OpenWISP Firmware) que son lanzadas con una licencia de código abierto, por esta razón se puede controlar a cualquier dispositivo que contenga una tarjeta Atheros Wi-Fi compatible con el firmware OpenWRT (Lista de dispositivos compatibles en el sitio web de OpenWRT). [3]

##### 2.1.2 OpenWISP2 Firmware

Es un paquete de OpenWRT basado en scripts de Shell que brinda un Daemon que en cierto espacio de tiempo recupera la configuración de WiFi, redes, capa 2 de tráfico modelado, openvpn y cronjobs del OpenWISP

Controller, además, ofrece una GUI web para la configuración de parámetros básicos de red, y administración de esta. [4]

### **2.1.3 OpenWISP2 Monitoring**

La característica principal de este módulo es la de ubicar geográficamente la topología de red, de cada uno de los dispositivos que la contengan, y proporcionar un mejor seguimiento de estos en caso de fallas o anomalías. [5]

## **2.2 Ansible**

Ansible es una herramienta de automatización de TI. Puede administrar y configurar sistemas, instalación o implementación de software y demás tareas TI mucho más avanzadas, como implementaciones continuas o actualizaciones. Su simplicidad y facilidad de uso, además de tener un fuerte enfoque en seguridad y confiabilidad y su uso de OpenSSH para el transporte, hacen de Ansible una herramienta muy útil, con un lenguaje que está diseñado para ser muy amigable y entendible para ser auditado. Ansible está descentralizado, se basa en sus credenciales de sistema operativo para controlar el acceso a máquinas remotas, manejándolas de una manera sin agente, por esta razón Ansible es apropiado para administrar cualquier tipo de entorno, sea empresarial con miles de instancias o pequeñas configuraciones con instancias de menor cantidad. [6]

## **2.3 Ubuntu Server**

Es un sistema Operativo de Código abierto, es la versión para servidores Ubuntu y proporcionar servicios a los clientes a través de la red, utiliza un entorno por línea de comandos (También se puede instalar un entorno gráfico) para optimizar los recursos y sólo instalar los software necesarios para proporcionar los servicios en la red, esto ayudara a la seguridad del servidor, además de contar con varias características adicionales , como mecanismos de autenticación y seguridad (firewall), servicios de correo , de archivos , de infraestructura de red (herramientas de respaldo, DNS, DHCP, VPN) , y servidor para aplicaciones web (Apache). [7]

Ubuntu server Soporta varios lenguajes de programación (PHP, Perl, Java, Ruby, Python) y administradores o gestores de bases de datos (MySQL, DB2, Oracle, PostgreSQL).

## 2.4 Debian 9

Es un sistema operativo de código libre, caracterizándose por soportar múltiples arquitecturas (alpha, amd64, armel, hppa, i386, ia64, mips, mipsel, powerpc, s390, y sparc) y poder correr en varios kernels GNU Hurd y FreeBSD, además de Linux y con la utilidad de Bootstrap, haciendo difícil encontrar algún dispositivo que no corra en Debian. Posee una gran cantidad de softwares (51000 softwares distintos), un sistema estable, sin reinicios o actualizaciones que puedan llegar a colapsar el servidor, además ofrece una alta seguridad en su sistema, la disponibilidad del código fuente permite que la seguridad de Debian se evalúe de forma abierta, con lo que se consigue eludir que se instalen modelos pobres de seguridad. [8]

## 2.5 Cable Ethernet

Es uno de los elementos más importantes de una red, sean redes grandes de considerables distancias, así como redes pequeñas o locales. Es el tipo de cable que habitualmente se utiliza para interconectar todos los dispositivos que conforman una red, incluyendo impresoras, discos externos, routers, scanners, switches, y las propias computadoras, es el encargado de transportar todos los datos que usualmente transitan en una red. Un cable Ethernet estándar es un cable UTP 8 Categoría 5 que en su interior aloja 4 pares de hilos, el secreto de su velocidad es el de tener recubierto cada uno de sus hilos con una malla metálica para evitar las interferencias. [9]



**Figura 2.2 Cable Ethernet**

## 2.6 Router Gigabit Inalámbrico TP Link TL – WR1043ND

Equipado con la última tecnología en lo que es conexión a red doméstica, como en casa u oficinas, ofrece puertos Gigabit Ethernet para conexiones con cable y obtener una alta velocidad, asegurándose que los usuarios puedan disfrutar o interactuar con distintas aplicaciones sin sufrir interrupciones, cuenta además con un puerto USB que le sirve de almacenamiento permitiendo así que los usuarios puedan acceder a dicha información, al igual que puede ser utilizada por servidores FTP o de impresión, causando que las impresiones sean más flexibles.

El router cumple con la norma IEEE 802.11n, permitiendo que los usuarios puedan crear redes inalámbricas y lograr velocidades de 18X y 6X, tiene un control de ancho de banda basado en ip, que le permitirá al administrador determinar la cantidad de ancho de banda en cada dispositivo conectado a la red. Su configuración es fácil y rápida mediante un interfaz de usuario agradable para el usuario. [10]



Figura 2.3 Router TL – WR1043ND.

## 2.7 Router Cisco Linksys E4200

Contiene múltiples características que lo convierten en un router de gama alta de la marca, una de ellas son sus interfaces WAN y LAN Gigabit, su doble banda

simultánea de 2.4 GHz con velocidades de hasta 300 Mbps y 5 GHz con una velocidad de 450 Mbps, además de contar con un puerto USB para almacenamiento externo que puede ser usado tanto para los usuarios o para los servidores que estén conectados en la red.

Además, posee una memoria Flash de 16 MB, que lo hace óptimo para la instalación de firmware de mayor tamaño, tiene un largo alcance en su cobertura, su diseño es moderno y compacto, cuenta también con un software llamado “Cisco Connect”, que provee una ayuda a los usuarios para crear y administrar su propia red LAN local o doméstica. [11].



**Figura 2.4 Cisco Cisco E4200.**

## **2.8 Switch TP Link TL – SF1008D**

El switch Fast Ethernet TL-SF1008D, opera con 8 puertos RJ45 a 10/100 Mbps con la tecnología MDI/MDIX automático, dando como ventaja el despreocuparse por el tipo de cable que se va a conectar con los puertos, sea cable cruzado o puertos tipo uplink. Cuenta con una arquitectura que permite un switching sin bloqueos de reenvío o filtrado de paquetes para obtener un rendimiento óptimo a la mayor velocidad. Dispone de un soporte para el control de flujo IEEE 802.3x en modo full Duplex y Back-pressure en modo Half Duplex, además cuenta con 10 K de jumbo frame, obteniendo un alto rendimiento en las transferencias de archivos de gran tamaño y de esta manera aliviar la congestión del tráfico, como el aumentar la fiabilidad de funcionamiento con un ahorro del 60% de energía eléctrica, lo que lo convierte en una solución ecológica para la red de cualquier empresa. [12]



**Figura 2.5 Switch TL – SF1008D.**

### **2.9 Adaptador TP-Link USB 3.0 a Ethernet Gigabit**

El adaptador USB 3.0 a Ethernet Gigabit provee una solución rápida al proporcionar de conectividad Gigabit Ethernet a su ordenador sea de mesa o portátil a través de un puerto USB 3.0, también compatibles con los estándares anteriores USB 2.0/1.1, transfiriendo archivos a gran velocidad. Además, destaca su alta compatibilidad con varios sistemas operativos Windows (XP/vista/7/8/8.1), Mac OS X (10.9/10.10) y Linux OS. [13]



**Figura 2.6 Adaptador USB a Ethernet.**

## CAPÍTULO 3

### 3. IMPLEMENTACIÓN Y DESARROLLO.

La solución al problema descrito en los párrafos anteriores consiste en la implementación de un software gratuito con licencia de libre uso y compatible en la gran parte de las distribuciones del sistema operativo Linux el cual funcione como controlador de los APs con los que cuenta la facultad de la FIEC.

Este software controlador permitirá tomar el control de cada AP de la facultad siempre y cuando se asocien a él y de esta manera el software podrá modificar agregar o eliminar configuraciones en los APs.

Para que los APs se asocien al software controlador, estos deben de ejecutar el firmware de Openwrt en su interior, este firmware está basado en Linux brindando mayor libertad al momento de instalar los paquetes necesarios para establecer la comunicación con el software controlador.

Todas las configuraciones dentro del AP se pueden manejar desde el software controlador, por ejemplo, crear una red wifi, poner un SSID que identifique a la red, colocarle seguridad a la red mediante una contraseña y editar el canal de transmisión, así también como configurar el nombre del dispositivo su zona horaria configurar sus interfaces de red y agregar mapas geográficos previamente diseñados de la ubicación de cada AP, logrando todo ello mediante el software Openwisp2 Controller.

#### 3.1 Instalación del sistema operativo.

El software controlador se ejecuta en el sistema operativo Linux, utilizando la distribución de Ubuntu server versión 16.04.3, siendo estable y compatible con el controlador.

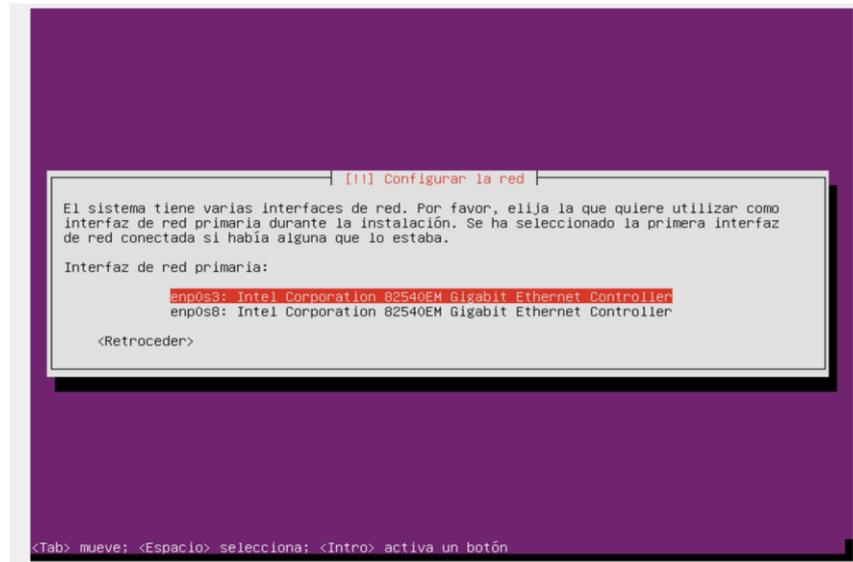
Los requisitos mínimos para la instalación Ubuntu server 16.04.3 son los siguientes:

Memoria RAM: 1 GB (recomendado 2 GB).

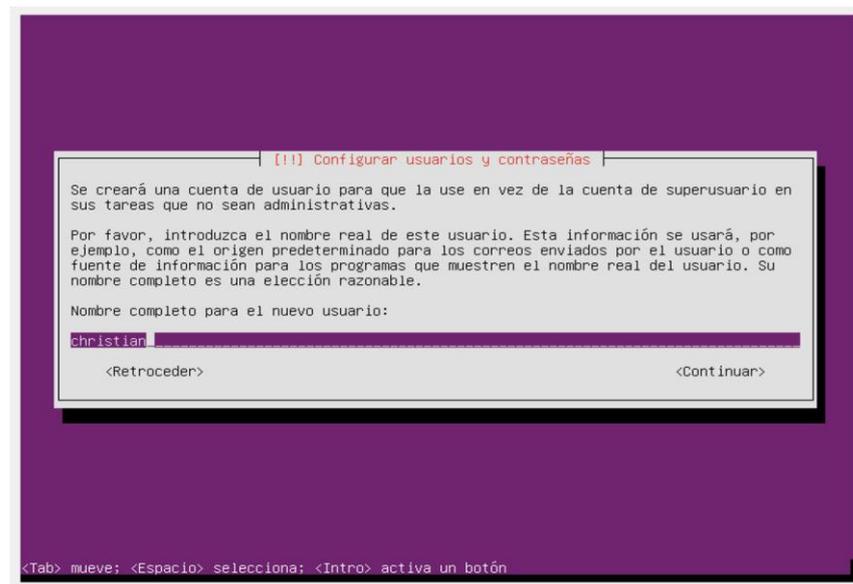
Procesador: Dual Core.

Disco Duro: 16 GB.

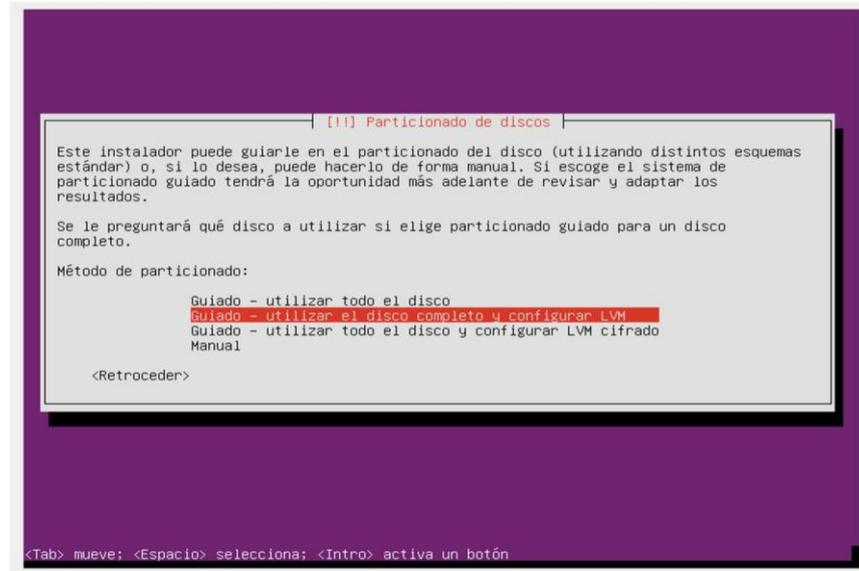
En primer lugar, se debe descargar el archivo.iso de la página oficial de Ubuntu, luego se procede a la instalación de Ubuntu server, donde realizamos algunas configuraciones a medida que se efectúa la instalación.



**Figura 3.1 Configuración de las interfaces de redes.**



**Figura 3.2 Creación de usuarios**



**Figura 3.3 Crear las particiones del disco.**

Una vez instalado Ubuntu server se obtendrá la siguiente pantalla, donde se puede iniciar sesión con el usuario y contraseña previamente creada.



**Figura 3.4 Interfaz de lista de comandos.**

### **3.1.1 Instalación de la interfaz gráfica de Ubuntu server.**

La distribución de Ubuntu server se maneja por la interfaz de línea de comandos, pero también se puede instalar la interfaz gráfica para un mejor entorno de trabajo mediante los siguientes comandos.

```
$ sudo apt-get update  
$ sudo apt-get upgrade  
$ sudo apt-get install ubuntu-desktop
```

Una vez aplicados los comandos anteriores se realiza un *reboot* al sistema y al reiniciar la maquina se obtiene la interfaz gráfica de Ubuntu Server.

### 3.2 Instalación de Openwisp2 Controller.

El software controlador que se va a utilizar es un software totalmente gratuito llamado Openwisp2 Controller el cual ayudara a gestionar y administrar todas las configuraciones que se realicen en los APs de la FIEC de tal manera que este tenga un control sobre estos, se eligió este software, ya que cuenta con muchas características relevantes como se observa en la Figura 3.5 entre ellas las de soportar dispositivos de diferentes marcas.

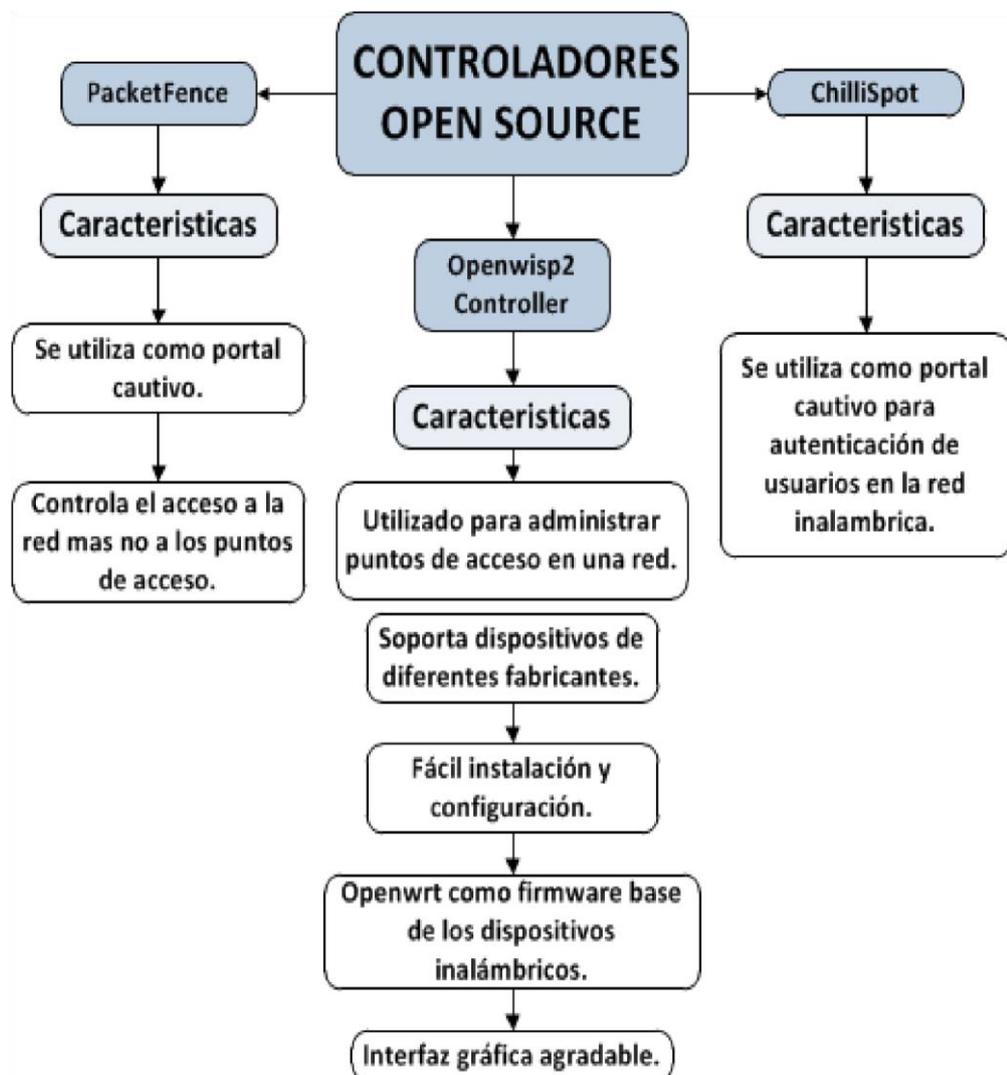


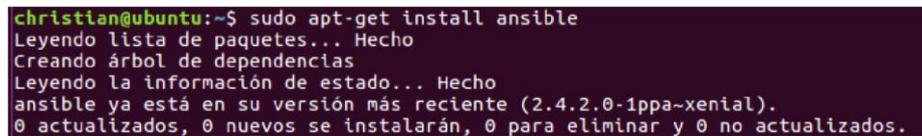
Figura 3.5 Controladores Open Source.

Para la instalación del controlador se debe de hacer uso de las siguientes herramientas.

### 3.2.1 Instalación de Ansible.

El primer lugar se debe instalar *Ansible versión 2.4* en la máquina local que se ejecuta en el sistema operativo Ubuntu, ya que por medio de este software configuramos el servidor que va a contener el controlador, el único requisito para instalar *Ansible* es contar con *Python 2 versiones 2.6 o 2.7* o *Python 3 versiones 3.5* o posteriores en la máquina local, verificado esto, se procede a colocar los siguientes comandos.

```
$ sudo apt-get update
$ sudo apt-get install software-properties-common
$ sudo apt-add-repository ppa:ansible/ansible
$ sudo apt-get update
$ sudo apt-get install Ansible
```



```
christian@ubuntu:~$ sudo apt-get install ansible
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
ansible ya está en su versión más reciente (2.4.2.0-1ppa~xenial).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

Figura 3.6 Instalación de Ansible.

Luego de esto se necesita instalar el software *git* mediante el siguiente comando.

```
$ sudo apt-get install git
```



```
christian@ubuntu:~$ sudo apt-get install git
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
git ya está en su versión más reciente (1:2.7.4-0ubuntu1.3).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

Figura 3.7 Instalación de Git

### 3.2.2 Instalando el rol Openwisp2.

Una vez instalado *Ansible* se procede a instalar el rol *openwisp2* el cual ayuda a cargar automáticamente archivos con tareas, plantillas, variables por defecto y configuraciones específicas a una determinada estructura, para la instalación de este rol hacemos uso de la herramienta *ansible-galaxy* que se instaló automáticamente al momento de instalar *Ansible*, se aplica el siguiente comando para instalar el rol.

```
$ ansible-galaxy install openwisp.openwisp2
```

```
christian@ubuntu:~$ ansible-galaxy install openwisp.openwisp2
- downloading role 'openwisp2', owned by openwisp
- downloading role from https://github.com/openwisp/ansible-openwisp2/archive/0.7.3.tar.gz
- extracting openwisp.openwisp2 to /home/christian/.ansible/roles/openwisp.openwisp2
- openwisp.openwisp2 (0.7.3) was installed successfully
- adding dependency: Stouts.postfix (origin/develop)
- extracting Stouts.postfix to /home/christian/.ansible/roles/Stouts.postfix
- Stouts.postfix (origin/develop) was installed successfully
```

Figura 3.8 Instalación de openwisp.openwisp2.

### 3.2.3 Directorio de Trabajo.

Luego de instalar el rol *openwisp2* se crea un directorio de trabajo en la máquina local donde se van a guardar todas las configuraciones de *openwisp2*, por tal motivo se establece una carpeta en el directorio raíz con el nombre de *openwisp2-ansible-playbook* y luego se accede a ella mediante los siguientes comandos.

```
$ mkdir ~/openwisp2-ansible-playbook
```

```
$ cd ~/openwisp2-ansible-playbook
```

```
christian@ubuntu:~$ mkdir ~/openwisp2-ansible-playbook
christian@ubuntu:~$ cd ~/openwisp2-ansible-playbook
christian@ubuntu:~/openwisp2-ansible-playbook$ █
```

Figura 3.9 Creación del directorio de trabajo.

### 3.2.4 Archivo de Inventario.

Dentro de la carpeta previamente definida se creará un archivo de inventario, donde se va a definir el servidor que contendrá el software controlador, este archivo se va a crear con el nombre `hosts` y va a contener lo siguiente.

```
$ vi hosts [openwisp2] openwisp2.midominio.com
```

Se debe reemplazar `openwisp2.midominio.com` por el hostname o dirección ip del servidor, así el archivo queda de la siguiente manera.



```
christian@ubuntu: ~/openwisp2-ansible-playbook
[openwisp2]
192.168.10.200
~
~
~
~
~
~
~
~
```

*Figura 3.10 Creación del archivo hosts.*

### 3.2.5 Creación del archivo Playbook.

A parte del archivo `hosts` se debe crear otro archivo llamado `playbook.yml` dentro de la carpeta `openwisp2-ansible-playbook`, donde este archivo permitirá configurar el controlador Openwisp2, brindando roles y variables de la siguiente manera.

```
$ vi Playbook.yml
hosts: openwisp2
    become: "{{ become | default('yes') }}"
roles:
openwisp.openwisp2 vars:
openwisp2_default_from_email: "prueba.espol.edu.ec"
```

```
christian@ubuntu: ~/openwisp2-ansible-playbook
- hosts: openwisp2
  become: "{{ become | default('yes') }}"
  roles:
    - openwisp.openwisp2
  vars:
    openwisp2_default_from_email: "prueba.espol.edu.ec"
~
~
```

**Figura 3.11 Creación del archivo `playbook.yml`.**

### 3.2.6 Ejecutando el archivo `hosts` y `Playbook.yml`.

Por último se debe ejecutar el archivo `hosts` y `playbook.yml` para implementar Openwisp2 en el servidor efectuando el siguiente comando en la máquina local.

```
$ sudo ansible-playbook -i hosts playbook.yml -u <user> -k --become -k
```

En donde `hosts` y `playbook.yml` son los archivos creados previamente y `user` es el nombre de usuario del servidor donde se va instalar y configurar Openwisp2 Controller.

```
christian@ubuntu:~/openwisp2-ansible-playbook$ ansible-playbook -i hosts playbook.yml -u christian -k --become -K
SSH password:
SUDO password[defaults to SSH password]:

PLAY [openwisp2] *****

TASK [Gathering Facts] *****
ok: [192.168.10.200]

TASK [Stouts.postfix : include_tasks] *****
included: /home/christian/.ansible/roles/Stouts.postfix/tasks/postfix.yml for 192.168.10.200

TASK [Stouts.postfix : include_tasks] *****
included: /home/christian/.ansible/roles/Stouts.postfix/tasks/install.deb.yml for 192.168.10.200

TASK [Stouts.postfix : Install requirements (Debian)] *****
```

**Figura 3.12 Ejecución de los archivos `hosts` y `playbook.yml`.**

```
christian@ubuntu: ~/openwisp2-ansible-playbook
skipping: [192.168.10.200]

TASK [openwisp.openwisp2 : Install clearsessions cronjob] *****
changed: [192.168.10.200]

TASK [openwisp.openwisp2 : Change superuser password hint] *****
ok: [192.168.10.200] => {
  "msg": "Change your admin password at https://192.168.10.200/admin/password_change/"
}

RUNNING HANDLER [openwisp.openwisp2 : reload systemd] *****
changed: [192.168.10.200]

RUNNING HANDLER [openwisp.openwisp2 : reload supervisor] *****
changed: [192.168.10.200]

RUNNING HANDLER [openwisp.openwisp2 : restart nginx] *****
changed: [192.168.10.200]

PLAY RECAP *****
192.168.10.200 : ok=58  changed=38  unreachable=0  failed=0

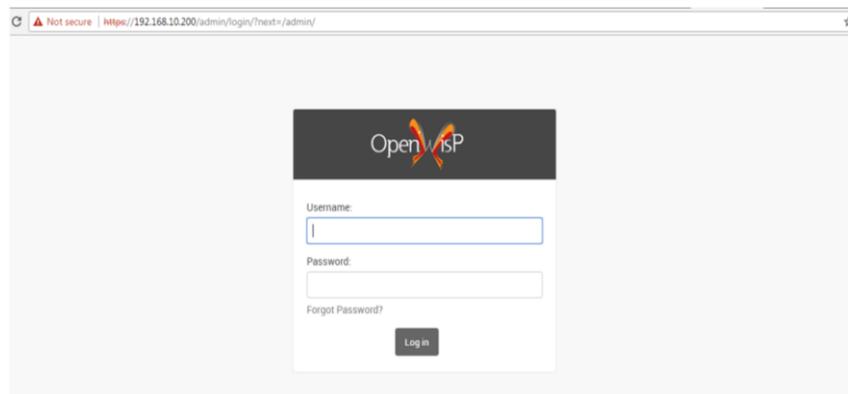
christian@ubuntu:~/openwisp2-ansible-playbook$
```

**Figura 3.13 Instalación y configuración de Openwisp2 Controller.**

### 3.3 Openwisp2 Controller.

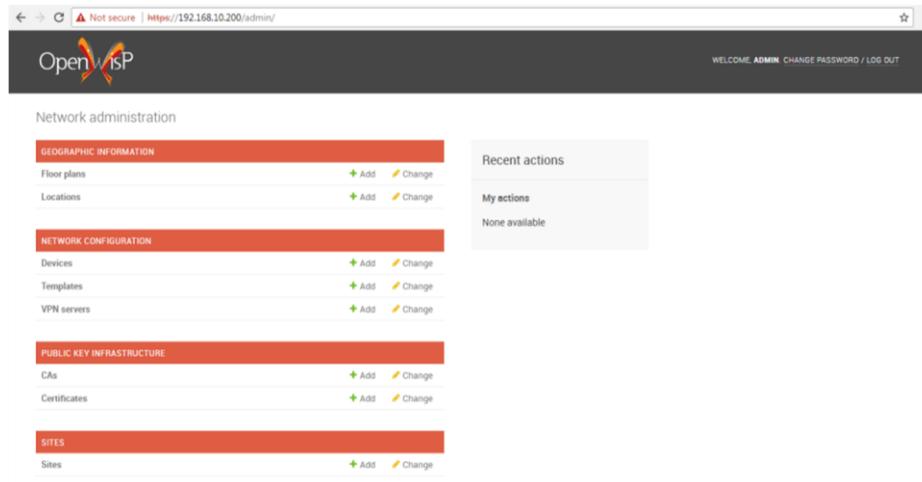
Una vez instalados los componentes y configuraciones del controlador se procede a ingresar al navegador web en la máquina local y entrar a la dirección del controlador mediante la dirección ip colocada en el archivo hosts.

<https://192.168.10.200/admin>



**Figura 3.14 Inicio de Sesión de Openwisp2 Controller.**

Luego iniciar sesión mediante el username: admin y el password: admin que son las credenciales que vienen por defecto para entrar al menú principal del controlador, no olvidar cambiar las credenciales en el menú principal para brindar una mayor seguridad, evitando el ingreso de cualquier intruso.



**Figura 3.15 Openwisp2 Controller.**

### **3.4 Instalación de Openwrt.**

Para que los APs se asocien al controlador se debe de instalar Openwrt como firmware base del dispositivo, para esto el AP debe cumplir con requisitos mínimos.

Memoria RAM: 32MB.

Memoria FLASH: 8MB.

La memoria flash es la más importante debido a cantidad paquetes que se van a necesitar instalar, por esta razón los dispositivos con memoria flash menor a 8MB no son los más apropiados para este tipo de proyectos. Para el proyecto se va a utilizar un Router inalámbrico marca TP-LINK TL-WR1043ND versión 5.0 el cual cuenta con memoria flash de 16 GB y el Router inalámbrico Cisco Linksys E4200 versión 1.0 que también cuenta con 16GB de memoria flash, en ambos routers se instalara Openwrt.

Para instalar Openwrt en el dispositivo se tiene que ir a la página oficial de Openwrt y buscar el modelo del dispositivo en la sección de Supported Devices, para este caso vamos a instalar Openwrt en el router TP-LINK TL-WR1043ND, el router es versión 5.0 pero la versión 3.0 de Openwrt es compatible entonces se procede a descargar dicha versión de firmware.

Filter: Current OpenWrt release >

Filtered by model ~wr1043

Show all (remove filter/sort)

#	Brand	Model	Versions	Current Release	Device Page	Device Techdata
1	TP-Link	TL-WR1043ND	v3.x	15.05.1	tl-wr1043nd	<a href="#">View/Edit data</a>
2	TP-Link	TL-WR1043ND	v1.x	15.05.1	tl-wr1043nd	<a href="#">View/Edit data</a>
3	TP-Link	TL-WR1043ND	v2.x	15.05.1	tl-wr1043nd	<a href="#">View/Edit data</a>

**Figura 3.16** Página oficial de Openwrt.

Luego entrar a la pestaña de View/Edit data y dar clic en Firmware Openwrt Install URL y automáticamente se comenzara a descargar archivo .bin del firmware para el dispositivo.

Button count: 2  
 Power Supply: 12 VDC, 1.5 A  
 Device Page: [tl-wr1043nd](#)  
 WikiDevi URL: [https://wikidevi.com/wiki/TP-LINK\\_TL-WR1043ND\\_v3.x](https://wikidevi.com/wiki/TP-LINK_TL-WR1043ND_v3.x)  
 OEM Device Homepage URL: [http://www.tp-link.com/en/download/TL-WR1043ND\\_V3.html](http://www.tp-link.com/en/download/TL-WR1043ND_V3.html)  
 Firmware OEM Stock URL: [http://www.tp-link.com/res/download/TL-WR1043ND\\_V3\\_150514.zip](http://www.tp-link.com/res/download/TL-WR1043ND_V3_150514.zip)  
 Firmware OpenWrt Install URL: <https://downloads.openwrt.org/snapshots/trunk/ar71xx/generic/openwrt-ar71xx-generic-tl-wr1043nd-v3-squashfs-factory.bin>  
 Firmware OpenWrt Upgrade URL: <https://downloads.openwrt.org/snapshots/trunk/ar71xx/generic/openwrt-ar71xx-generic-tl-wr1043nd-v3-squashfs-sysupgrade.bin>

Picture:



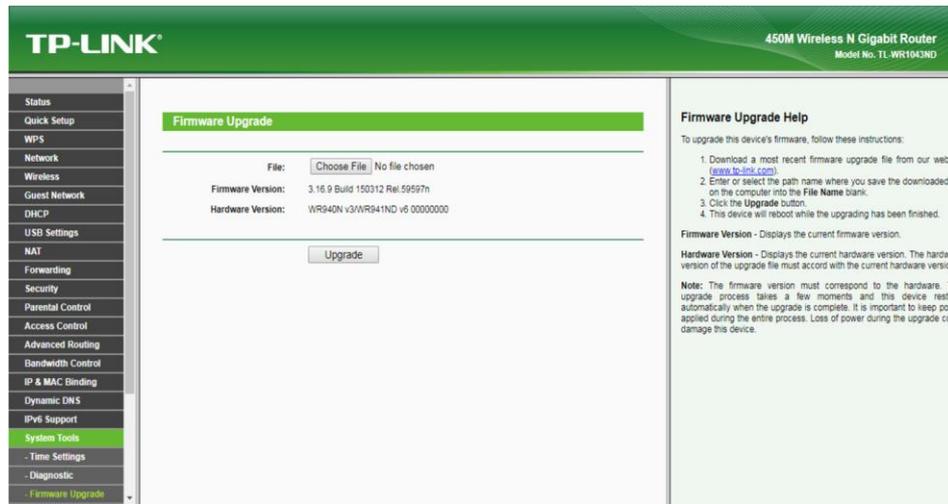
**Figura 3.17** Firmware de Openwrt para el Router WR1043ND.

Una vez descargado el archivo .bin conectamos un extremo del cable a cualquier puerto LAN del Router y el otro extremo al puerto Ethernet de una PC, luego el Router va asignar una dirección ip por dhcp a la PC, se entra al cmd de la PC y colocar el comando ipconfig, donde se puede observar la puerta de enlace del Router, ya que mediante esa ip se va a entrar al panel de administración, para este caso la dirección ip 192.168.10.1 es la puerta de enlace.

```
Ethernet adapter Local Area Connection:
Connection-specific DNS Suffix . . . : lan
IPv6 Address. . . . . : fd23:7f4a:e8a9::72e
IPv6 Address. . . . . : fd23:7f4a:e8a9:0:c91f:861:3c76:f3e3
Temporary IPv6 Address. . . . . : fd23:7f4a:e8a9:0:b9ff:eb28:544b:f779
Link-local IPv6 Address . . . . . : fe80::c91f:861:3c76-f3e3%10
IPv4 Address. . . . . : 192.168.10.235
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.10.1
```

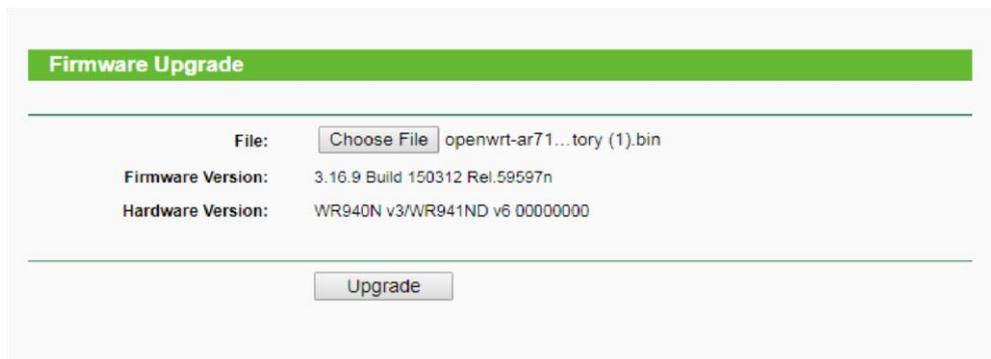
**Figura 3.18** Símbolo de sistema.

Se va a ingresar por medio de la dirección ip 192.168.10.1 al panel de administración del dispositivo y luego ir a la pestaña de System Tools.



**Figura 3.19 Panel de administración de Router WR1043ND.**

Luego ir a las pestañas Firmware Upgrade, donde se debe buscar el archivo .bin previamente descargado y dar clic en Upgrade.



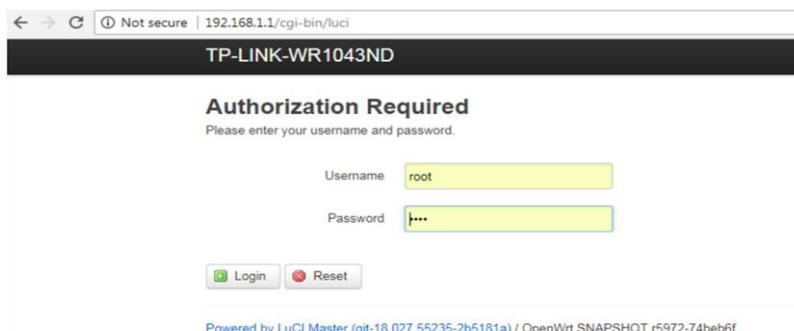
**Figura 3.20 Instalación del Firmware Openwrt.**

Concluido el proceso de Upgrade se procede a ingresar a la cmd para verificar la puerta de enlace del router con el nuevo firmware openwrt.

```
Ethernet adapter Local Area Connection:
Connection-specific DNS Suffix . . . : lan
IPv6 Address . . . . . : fd23:7f4a:e8a9::72e
IPv6 Address . . . . . : fd30:8f5a:34d3::849
IPv6 Address . . . . . : fd30:8f5a:34d3:0:c91f:861:3c76:f3e3
Temporary IPv6 Address . . . . . : fd30:8f5a:34d3:0:6c63:6152:b85c:272f
Link-local IPv6 Address . . . . . : fe80::c91f:861:3c76:f3e3%10
IPv4 Address . . . . . : 192.168.1.192
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.1.1
```

**Figura 3.21 Símbolo de sistema Openwrt.**

Se ingresa nuevamente al panel de administración del router, pero ahora mediante la ip 192.168.1.1, siendo la nueva puerta de enlace del router con la interfaz de Openwrt, root para el usuario y root para la contraseña son los valores por defecto para ingresar al router, no olvidar cambiar las credenciales por defecto en el menú principal de Openwrt para maximizar la seguridad, evitando el ingreso de cualquier intruso.

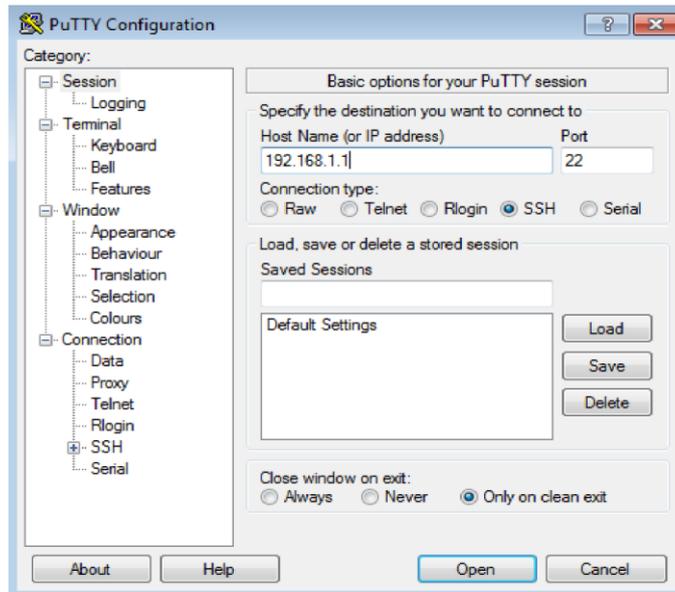


**Figura 3.22 Interfaz de Openwrt router WR1043ND.**

### 3.5 Instalación de openwisp-config-openssl.

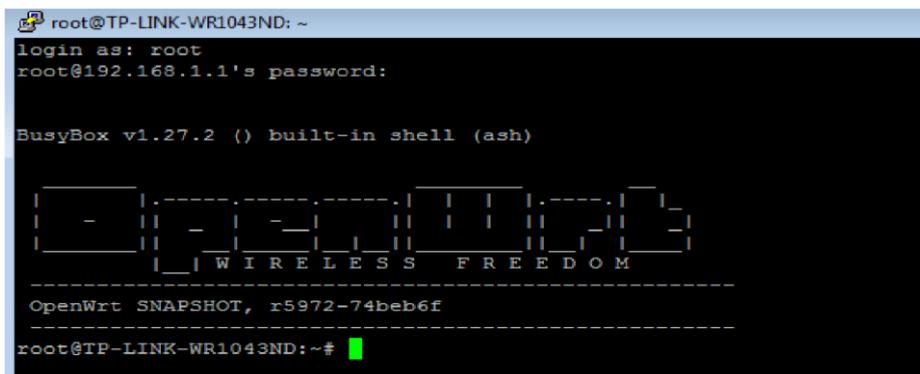
A más de tener instalado Openwrt como firmware base del router se necesita instalar un paquete adicional llamado openwisp-config-openssl, ayudando a la conexión entre el router inalámbrico y el controlador Openwisp2 Controller, para ello se tiene que seguir los siguientes pasos.

Mediante el putty se ingresa al router por ssh colocando la dirección ip 192.168.1.1, se mostrará una advertencia al ingresar y dar clic en sí.



**Figura 3.23 Putty Configuration.**

Se ingresa en el usuario root y en la contraseña root que son los valores por defecto para acceder al router por ssh.

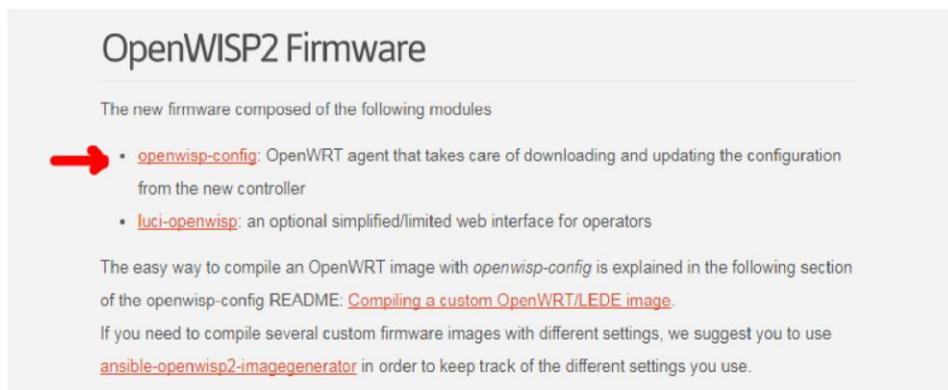


**Figura 3.24 Ventana de ingreso al router por ssh.**

Una vez que se logre ingresar al router vía ssh primero se debe actualizar la lista de paquetes disponibles en el router mediante el siguiente comando.

```
root@TP-LINK-WR1043ND:~#opkg update
```

Luego ingresar a la página oficial de Openwisp y a la pestaña de openwisp-config en la sección de Openwips2 Firmware.



**Figura 3.25** *Página oficial de Openwisp.*

Ingresar a la pestaña [downloads.openwisp.org](https://downloads.openwisp.org), donde se encuentran los paquetes necesarios para asociar el router inalámbrico con el controlador y obtener la siguiente pantalla.

## Index of /openwisp-config

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>		-	
<a href="#">latest/</a>	2017-07-13 14:21	-	
<a href="#">0.4.5-17-gf61910c/</a>	2017-07-13 14:21	-	
<a href="#">0.4.5-15-gd6eb1a3/</a>	2017-06-01 19:04	-	
<a href="#">0.4.5-14-ga9d43c2/</a>	2017-05-15 14:40	-	
<a href="#">0.4.5-13-gba3ea80/</a>	2017-04-27 19:18	-	
<a href="#">0.4.5-13-g06199ab/</a>	2017-04-26 17:19	-	
<a href="#">0.4.5-12-g90d2e00/</a>	2017-04-19 15:32	-	
<a href="#">0.4.5-13-g760c0f4/</a>	2017-04-14 17:13	-	
<a href="#">0.4.5-12-gb87906d/</a>	2017-04-13 17:19	-	
<a href="#">0.4.5-11-gb364375/</a>	2017-04-07 10:59	-	
<a href="#">0.4.5-10-g47701f4/</a>	2017-03-22 16:25	-	
<a href="#">0.4.5-9-g5c4f214/</a>	2017-03-16 16:21	-	

**Figura 3.26** *downloads.openwisp.org.*

Se escoge la pestaña [latest/](#) que es la última versión del paquete que se va a instalar, aparecerá una ventana en donde se va a tener 5 opciones de archivos ipk que se pueden descargar, para este caso se va a utilizar el paquete `openwisp-config-openssl` debido a que este no da problemas en la instalación y trabaja muy bien en cualquier tipo de router inalámbrico sin importar el fabricante, dar clic derecho y copiar la dirección del enlace.

## Index of /openwisp-config/latest

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>		-	
<a href="#">openwisp-config-polarssl_0.4.6a-1_all.ipk</a>	2017-07-13 14:21	11K	
<a href="#">openwisp-config-openssl_0.4.6a-1_all.ipk</a>	2017-07-13 14:21	11K	
<a href="#">openwisp-config-noss</a>		11K	Open link in new tab
<a href="#">openwisp-config-mbed</a>		57K	Open link in new window
<a href="#">openwisp-config-cyas</a>		57K	Open link in incognito window

Save link as...

Copy link address

Inspect Ctrl+Shift+I

**Figura 3.27 openwisp-config-openssl.**

Volver al putty y ejecutar el comando `opkg install URL` para descargar el paquete deseado, en la opción URL se debe ingresar la dirección de enlace previamente copiada, automáticamente se comenzará a descargar el paquete `openwispconfig-openssl` y sus dependencias como `ca-certificates` y `libopenssl`.

```
root@TP-LINK-WR1043ND:~#opkg install
http://downloads.openwisp.org/openwispconfig/latest/openwisp-config
openssl_0.4.6a-1_all.ipk
```

### 3.5.1 Configuración de `openwisp-config-openssl`.

Instalado el paquete `openwisp-config-openssl` se procede a configurarlo, ingresando a la siguiente ruta `/etc/config/` y abrir el archivo `Openwisp` con cualquier editor de texto y proceder a llenar los siguientes campos obligatorios para que exista conexión entre el router inalámbrico y el controlador.

```
root@TP-LINK-WR1043ND:~# cd /etc/config/ root@TP-LINK-
WR1043ND:/etc/config# vi openwisp
```

```

config controller 'http'
#option url 'https://openwisp2.mynetwork.com'
#option interval '120'
#option verify_ssl '1'
#option shared_secret ''
#option consistent_key '1'
#option mac_interface 'eth0'
#option merge_config '1'
#option test_config '1'
#option test_script '/usr/sbin/mytest'
option uuid ''
option key ''
list unmanaged 'system.@led'
list unmanaged 'network.loopback'
list unmanaged 'network.@switch'
list unmanaged 'network.@switch_vlan'
# curl options
#option connect_timeout '15'
#option max_time '30'
#option capath '/etc/ssl/certs'
#option cacert '/etc/ssl/certs/ca-certificates.crt'
# hooks
#option pre_reload_hook '/usr/sbin/my_pre_reload_hook'
- /etc/config/openwisp 23/26 88%

```

**Figura 3.28** Archivo openwisp.

El primer campo option url es el más importante, ya que ahí se debe colocar la url del controlador Openwisp2 Controller, se debe quitar el # a la línea para que no sea tomada como comentario y luego remplazar **https://openwisp2.mynetwork.com** con la url del controlador que en este caso sería **https://192.168.10.200/**.

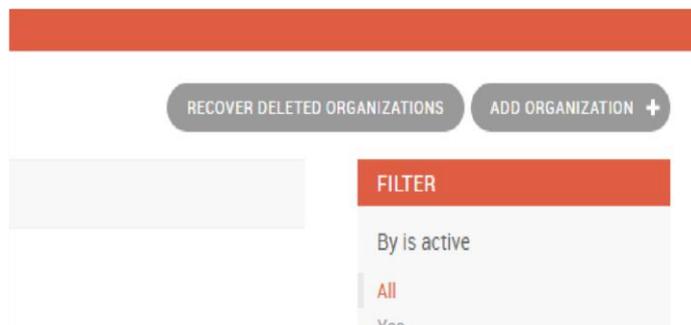
```

config controller 'http'
option url 'https://192.168.10.200/'

```

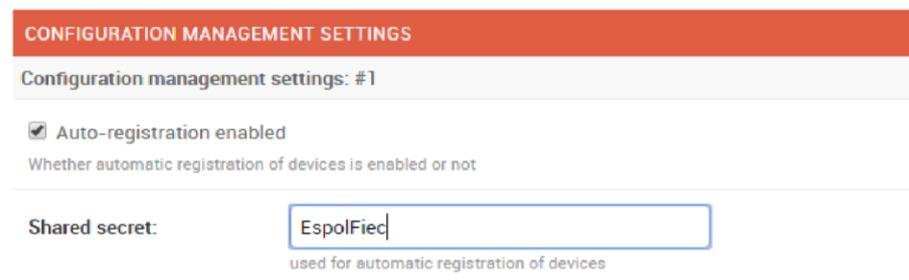
**Figura 3.29** Configuración de option url.

El campo option shared\_secret debe ser llenado obligatoriamente, ya que por medio de este vamos a asociar el router inalámbrico con Openwisp2 Controller, este campo consiste en una clave compartida y es creada por el usuario dentro del controlador ingresando a la opción Organizations, dar clic en ADD ORGANIZATION.



**Figura 3.30 Creación de una Organización.**

Luego llenar los campos requeridos como el nombre de la organización, en la opción Shared secret escribir la clave compartida deseada, en este caso EspolFiec, dar clic en la opción Auto-registration enable, de no encontrarse marcada, permite que cuando un router inalámbrico tenga la misma clave compartida en el campo option shared\_secret el controlador lo registre automáticamente, por último, dar clic en guardar.



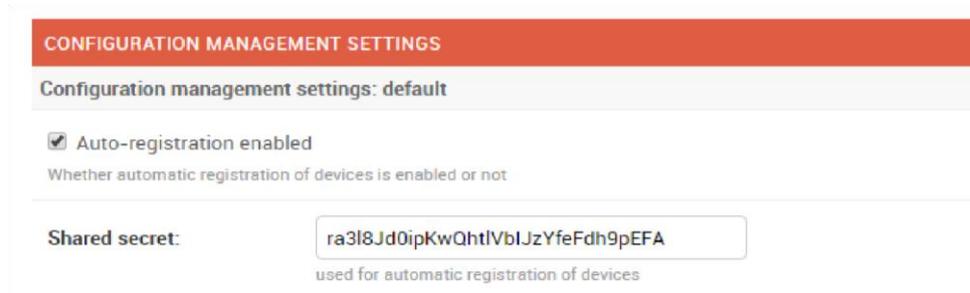
**Figura 3.31 Creación de Shared secret.**

Una vez creado el Shared secret volver al archivo openwisp y colocar dentro del campo option shared\_secret la clave compartida previamente creada, no olvidar quitar el # en este campo quedando lo siguiente.

```
option shared_secret 'EspolFiec'
```

**Figura 3.32 Campo option shared\_secret.**

Si no se desea crear una clave compartida puede colocar en el campo la clave por defecto ya creada en el controlador en la parte de Organizations en la opción default.



CONFIGURATION MANAGEMENT SETTINGS

Configuration management settings: default

Auto-registration enabled  
Whether automatic registration of devices is enabled or not

Shared secret:   
used for automatic registration of devices

**Figura 3.33 Shared secret por defecto.**

Por último se tiene los campos option uuid y option key, estos campos son únicos e identifican a los dispositivos de los demás, estos campos se llenan automáticamente al momento de que el router inalámbrico se asocie al controlador. Una vez llenado los campos necesarios el archivo openwisp debe quedar de la siguiente manera.

```
config controller 'http'  
  option url 'https://192.168.10.200/'  
  #option interval '120'  
  #option verify_ssl '1'  
  option shared_secret 'EspolFiec'  
  #option consistent_key '1'  
  #option mac_interface 'eth0'  
  #option merge_config '1'  
  #option test_config '1'  
  #option test_script '/usr/sbin/mytest'  
  option uuid ''  
  option key ''
```

**Figura 3.34 Archivo openwisp configurado.**

## CAPÍTULO 4

### 4. PRUEBAS Y RESULTADOS.

Se implementó un software controlador en Ubuntu-server el cual sea capaz de administrar routers inalámbricos con el fin de crear una red inalámbrica estable, segura, escalable y con un mismo SSID en todos los dispositivos, las pruebas se realizaron con routers inalámbricos marca CISCO Y TP-LINK con el firmware de Openwrt en su interior los cuales van a funcionar como APs, a continuación en la Figura 4.1 se muestra el diagrama de red utilizado para realizar las pruebas del controlador y verificar que esté funcionando correctamente.

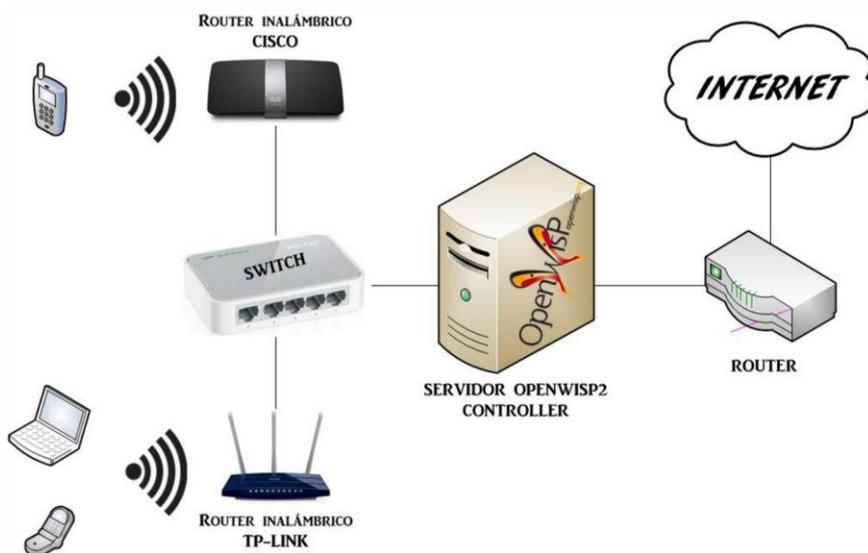


Figura 4.1 Diagrama de red Openwisp2 Controller.

#### 4.1 Pruebas de Registro.

Una vez realizada las configuraciones dentro de los dispositivos se debe colocar el siguiente comando para que Openwisp2 Controller comience a registrar los routers inalámbricos.

```
root@TP-LINK-WR1043ND:~# /etc/init.d/openwisp_config start
```

Primero se aplica el comando anterior al router TP-LINK TL-WR1043ND, luego se necesita observar los logs del dispositivo en donde se muestra como el dispositivo se registra automáticamente a Openwisp2 Controller, para ver los logs del dispositivo aplicamos el siguiente comando.

```
root@TP-LINK-WR1043ND:~# logread | grep openwisp
```

```
Wed Feb 14 06:37:33 2018 daemon.info openwisp: OpenNISP config agent started
Wed Feb 14 06:37:34 2018 daemon.info openwisp: Registering device...
Wed Feb 14 06:37:34 2018 daemon.info openwisp: New device registered successfully as 50:C7:BF:DC:F7:6C, id: b63942bfaae4288b114d7ca06227a2c
Wed Feb 14 06:37:38 2018 daemon.info openwisp: Local configuration outdated
Wed Feb 14 06:37:38 2018 daemon.info openwisp: Downloading configuration from controller...
Wed Feb 14 06:37:38 2018 daemon.info openwisp: Configuration downloaded, now applying it...
Wed Feb 14 06:37:38 2018 daemon.info openwisp: The following uci configs have been renamed: dhcp.cfg01411c, dropbear.cfg014dd4, firewall.cfg0692bd, firewall.cfg0792bd, firewall.cfg0c92bd, firewall.cfg0d92bd, firewall.cfg04ad58, firewall.cfg01c63d, firewall.cfg0892bd, firewall.cfg034c81, rpcd.cfg01f8be, network.cfg063777, network.cfg041ec7, network.cfg091ec7, system.system, ucitrack.cfg05ad54, ucitrack.cfg08cd75, ucitrack.cfg0e8036, ucitrack.cfg10822b, ucitrack.cfg0f0f89, ucitrack.cfg03b57b, ucitrack.cfg0d804c, ucitrack.cfg0c003c, ucitrack.cfg0102af, ucitrack.cfg088cc9, ucitrack.cfg04c845, ucitrack.cfg02e233, ucitrack.cfg0aa258, ucitrack.cfg06d057, ucitrack.cfg074dd4, ucitrack.cfg0be48a
Wed Feb 14 06:37:39 2018 daemon.info openwisp: Service dhcp has been reloaded via procd/ubus
Wed Feb 14 06:37:39 2018 daemon.info openwisp: Service dropbear has been reloaded via procd/ubus
Wed Feb 14 06:37:39 2018 daemon.info openwisp: Service firewall has been reloaded via procd/ubus
Wed Feb 14 06:37:39 2018 daemon.info openwisp: Service network has been reloaded via procd/ubus
Wed Feb 14 06:37:39 2018 daemon.info openwisp: Service openwisp has been reloaded via procd/ubus
Wed Feb 14 06:37:40 2018 daemon.info openwisp: Service rpcd has been reloaded via init.d script
Wed Feb 14 06:37:40 2018 daemon.info openwisp: Service system has been reloaded via procd/ubus
Wed Feb 14 06:37:40 2018 daemon.info openwisp: Service ucitrack has been reloaded via procd/ubus
Wed Feb 14 06:37:40 2018 daemon.info openwisp: Service wireless has been reloaded via procd/ubus
Wed Feb 14 06:37:41 2018 daemon.info openwisp: OpenNISP config agent received reload trigger
Wed Feb 14 06:37:45 2018 daemon.info openwisp: Testing configuration...
Wed Feb 14 06:37:45 2018 daemon.info openwisp: Configuration test succeeded
Wed Feb 14 06:37:45 2018 daemon.info openwisp: Configuration applied successfully
Wed Feb 14 06:37:46 2018 daemon.info openwisp: OpenNISP config agent started
root@50-C7-BF-DC-F7-6C:~#
```

Figura 4.2 Registro del dispositivo a Openwisp2 Controller.

Para verificar que el dispositivo se registró correctamente, se ingresa a la página web del controlador <https://192.168.10.200/> dando clic en la pestaña de Devices se encuentran los dispositivos consignados.

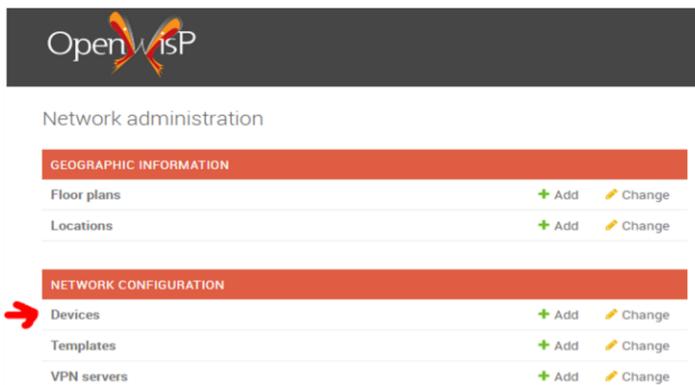


Figura 4.3 Pestaña Devices de Openwisp2 Controller.

Select device to change

Action: -----  0 of 1 selected

	NAME	ORGANIZATION	BACKEND	STATUS	LAST IP	CREATED	MODIFIED
→ <input type="checkbox"/>	50:C7:BFDC:F7:6C	Espol	OpenWRT/LEDE	running	192.168.10.3	17 Feb 2018, 4:40 p.m.	17 Feb 2018, 4:40 p.m.

Action: -----  0 of 1 selected

**Figura 4.4 Dispositivo TP-LINK TL-WR1043ND registrado.**

Dando clic en el dispositivo se puede verificar información de este como la dirección MAC, el modelo, el sistema operativo y aquí se puede cambiar el nombre del dispositivo para que se pueda distinguir con el resto al momento que se tengan muchos routers asociados al controlador, en el nombre se va a colocar TP-LINK WR1043ND.

Name:

---

Organization: Espol 👉 +

---

Mac address:   
primary mac address

---

UUID: b63942bfaee4288b114d7cd06227a2c

---

Key:   
unique device key

---

Model:   
device model and manufacturer

---

Operating system:   
operating system identifier

---

SOC / CPU:   
system on chip or CPU info

---

Created: 17 Feb 2018, 4:40 p.m.

---

Modified: 17 Feb 2018, 4:40 p.m.

**Figura 4.5 Información general del router TP-LINK TL-WR1043ND.**

Name:

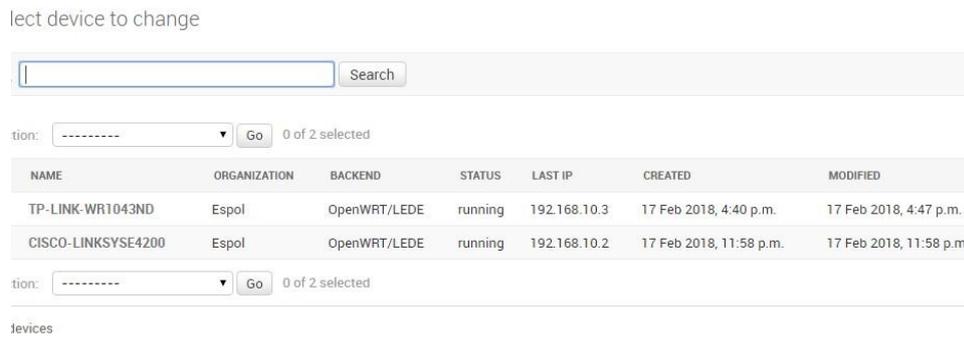
**Figura 4.6 Cambio de nombre al router TP-LINK TL-WR1043ND.**

Ahora es el turno de registrar el router Cisco Linksys E4200 se aplica el mismo comando anterior para registrar el dispositivo al controlador y luego se necesita ver los logs del dispositivo.

```
Sat Feb 3 19:36:54 2018 daemon.info openwisp: OpenWISP config agent started
Sat Feb 3 19:36:54 2018 daemon.info openwisp: Registering device...
Sat Feb 3 19:36:56 2018 daemon.info openwisp: New device registered successfully as CISCO-LINKSYS4200, id: 2697a33cfd094b182c1fc2b404a1fe3
Sat Feb 3 19:37:00 2018 daemon.info openwisp: Local configuration outdated
Sat Feb 3 19:37:00 2018 daemon.info openwisp: Downloading configuration from controller...
Sat Feb 3 19:37:00 2018 daemon.info openwisp: Configuration downloaded, now applying it...
Sat Feb 3 19:37:03 2018 daemon.info openwisp: Service network has been reloaded via procd/ubus
Sat Feb 3 19:37:03 2018 daemon.info openwisp: Service openwisp has been reloaded via procd/ubus
Sat Feb 3 19:37:03 2018 daemon.info openwisp: Service system has been reloaded via procd/ubus
Sat Feb 3 19:37:03 2018 daemon.info openwisp: Service wireless has been reloaded via procd/ubus
Sat Feb 3 19:37:03 2018 daemon.info openwisp: OpenWISP config agent received reload trigger
Sat Feb 3 19:37:08 2018 daemon.info openwisp: Testing configuration...
```

**Figura 4.7 Registro del dispositivo Cisco a Openwisp2 Controller.**

Para verificar que el dispositivo se registró correctamente se tiene que ir a la página web del controlador <https://192.168.10.200/> y en la parte de Devices verificar si se encuentra el dispositivo.



**Figura 4.8 Dispositivo Cisco Linksys E4200 registrado.**

## 4.2 Creación de plantillas.

Con los dispositivos registrados en el controlador lo siguiente que se va a realizar es la creación de plantillas, son el ítems más importante del controlador y ayudaran a la configuración de todos los dispositivo registrados sea de cualquier fabricante, facilitando la configuración de cada dispositivo, ya que no se tiene que configurar como por ejemplo sus direcciones de red mediante línea de comandos sino que se va hacer gráficamente mediante las plantillas de Openwisp2 Controller y así el usuario no necesita saber ningún lenguaje de programación o comandos de Linux avanzados para configurar dispositivos.

### 4.2.1 Plantilla Zona Horaria.

Se va a crear una plantilla para configurar la zona horaria en el dispositivo, esto permitirá que en el dispositivo se configure su zona horaria, también se puede habilitar esta plantilla para que al momento de que un dispositivo nuevo se registre esta plantilla se aplique automáticamente.

Para crear una plantilla ir a la pestaña Templates y elegir la opción ADD TEMPLATE y elegimos el nombre de la plantilla y seleccionamos la opción Enabled by default.



Figura 4.9 Agregando una plantilla.

Name:

Organization:   

Type:    
 template type, determines which features are available

Backend:    
 Select [netjsonconfig](#) backend

Tags:    
 A comma-separated list of template tags, may be used to ease auto configuration with specific settings (eg. 4G, mesh, WDS, VPN, ecc.)

Enabled by default   
 whether new configurations will have this template enabled by default

Figura 4.10 Configuración del nombre a la Plantilla.

Luego en la parte de abajo dar clic en la opción Configuration Menu y agregar la opción General, se desplegará un recuadro con varias opciones de rellenar y en la parte de timezone elegir la región para que de esta manera configurar la zona horaria, luego dar clic en guardar.

**Configuration Menu**

**General**

timezone:

---

ULA prefix:   
IPv6 Unique Local Address prefix

---

maintainer:

---

description:   
description and notes

---

Created: 17 Feb 2018, 5:23 p.m.

---

Modified: 17 Feb 2018, 5:23 p.m.

**Figura 4.11 Configuración de la Zona horaria.**

Para verificar que la plantilla se haya creado correctamente ir a la pestaña de Templates en el menú principal.

Select template to change RECOVER

Action:   0 of 1 selected

<input type="checkbox"/>	NAME	ORGANIZATION	TYPE	BACKEND	ENABLED BY DEFAULT	CREATED	MODIFIED
<input type="checkbox"/>	Zona Horaria	Espol	Generic	OpenWRT/LEDE	✔	17 Feb 2018, 5:28 p.m.	17 Feb 2018, 6:34 p.m.

Action:   0 of 1 selected

**Figura 4.12 Menú de plantillas.**

#### 4.2.2 Plantilla WIFI.

Se va a crear una plantilla llamada WIFI-TPLINK la cual va permitir crear y configurar una red inalámbrica dentro del dispositivo, lo primero que tenemos que hacer es ir a la pestaña de Configuration Menu y elegir la opción de Interfaces y radio.

Interfaces

Interface 1 Wireless interface ▾

name

type wireless ▾

MTU

auto start bring up interface on boot

wireless Access Point ▾

mode access point ▾

radio   
reference to one of the elements defined in the "radios" section

SSID

**Figura 4.13** Configuración Menú.

Name:

---

Organization: Espol ▾  

---

Type: Generic ▾  
template type, determines which features are available

---

Backend: OpenWRT/LEDE ▾  
Select [netjsonconfig](#) backend

---

Tags:   
A comma-separated list of template tags, may be used to ease auto configuration with specific settings (eg: 4G, mesh, WDS, VPN, ecc.)

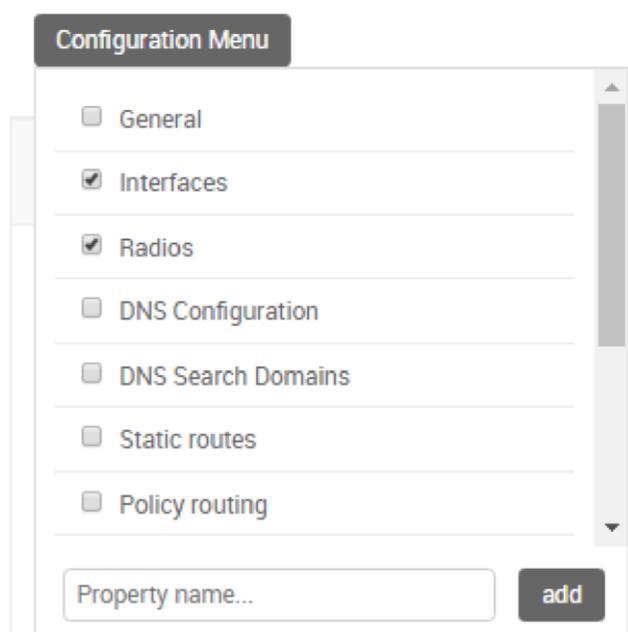
---

Enabled by default  
whether new configurations will have this template enabled by default

**Configuration:**

**Figura 4.14** Creación de una interfaz de red.

Dar clic en Add interface para crear una interfaz de red inalámbrica en el dispositivo, llenar los campos como el tipo de interfaz que será inalámbrica, el nombre de la interfaz que por lo general los dispositivos manejan nombres como interface wlan0 o wlan1, elegir el modo en que va trabajar el dispositivo que en este caso sería Access Point, elegir el radio que más adelante se va a definir y colocar el SSID deseado.



**Figura 4.15 Configuración de una interfaz de red.**

La opción hide SSID permite que la red inalámbrica no sea visible para este caso no la seleccionamos y en la opción network 1 llenar con el dato lan, en el tipo de seguridad elegir la deseada, en este caso se ha elegido WPA2/WPA PERSONAL y se ha colocado una clave sencilla compuesta de números para realizar las pruebas.

<input checked="" type="checkbox"/> WMM (802.11e)	enables WMM (802.11e) support; required for 802.11n support
MAC Filter	disable ▾ <small>specifies the mac filter policy, "disable" to disable the filter, "allow" to treat it as whitelist or "deny" to treat it as blacklist</small>
<b>Attached Networks</b>	
<small>override OpenWRT "network" config option of wifi-iface directive, will be automatically determined if left blank</small>	
network 1	lan
<b>Add network</b>	<b>Delete Last network</b>
Encryption	WPA2/WPA Personal ▾
encryption protocol	WPA2 Personal ▾
key	12345678
cipher	auto ▾

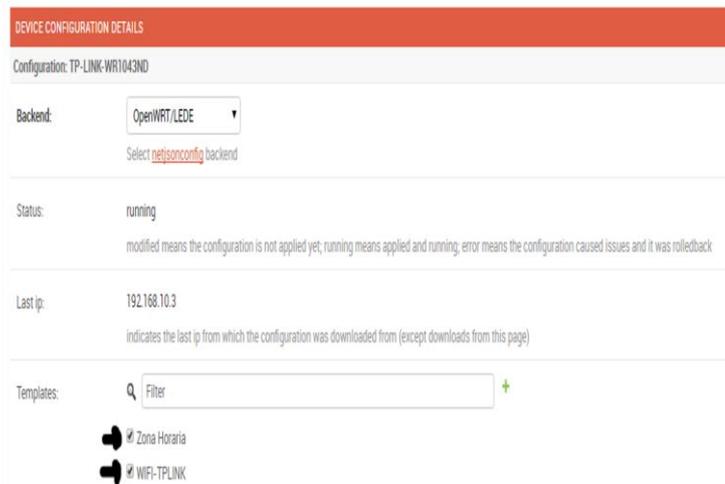
**Figura 4.16 Configuración de la seguridad de la red.**

Ahora se va a crear y configura el Radios para una red inalámbrica, elegir el tipo de Radio que soporta el dispositivo para el caso del dispositivo TPLINK TL-WR1043ND este trabaja con el estándar 802.11n con frecuencia de 2.4 GHz N, se elige el nombre del Radios que por lo general los dispositivos en esta apartado manejan el nombre de radio0 o radio1 dependiendo en qué tipo de frecuencia trabajen sea en 2.4 o 5 GHz, también se elige el protocolo que es el 802.11n, el tipo de driver del dispositivo que es el mac80211, elegimos el canal en que va a trabajar la red inalámbrica, este apartado es muy importante ya que se tiene que elegir un canal adecuado entre dispositivos cercanos para que no exista interferencia en la transmisión y recepción de datos, ya que si dos canales se traslapan pueden existir pérdidas de paquetes o a su vez la conexión suele ir lenta, por último elegir el ancho del canal y la potencia de transmisión de datos, elegir el país y luego dar clic en guardad para que la plantilla se genere.

Radios	
Radio 1	802.11n (2.4 GHz N) ▼
name	radio0
protocol	802.11n ▼
driver	mac80211 ▼
channel	11 ▼
channel width (mhz)	20 ▼
transmit power (dbm)	20
country	Ecuador ▼
hardware mode	11g ▼
<input type="checkbox"/> disabled	

**Figura 4.17 Configuración de Radios plantilla WIFI-TPLINK.**

Una vez creada la plantilla WIFI-TPLINK dirigirse al apartado Devices en el menú principal de Openwisp2 Controller y escoger el dispositivo TPLINK TL-WR1043ND al cual se le va aplicar la plantilla WIFI-TPLINK previamente creada, en la parte DEVICE CONFIGURATION DETAILS en la pestaña de Templates se encuentran las plantillas creadas y se selecciona WIFI-TPLINK para aplicarla al dispositivo, dar clic en guardar.



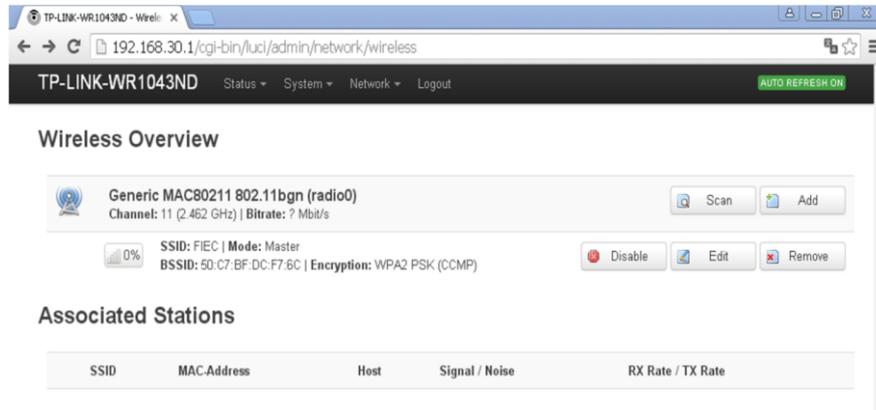
**Figura 4.18** Uso de la plantilla WIFI-TPLINK.

Para verificar si la plantilla se ha aplicado en el dispositivo es necesario ver los logs del dispositivo con el comando `logread | grep openwisp`.

```
root@TP-LINK-WR1043ND:~# logread | grep openwisp
Wed Feb 14 02:02:36 2018 daemon.info openwisp: Testing configuration...
Wed Feb 14 02:02:36 2018 daemon.info openwisp: Configuration test succeeded
Wed Feb 14 02:02:37 2018 daemon.info openwisp: Configuration applied successfully
root@TP-LINK-WR1043ND:~# logread | grep openwisp
Wed Feb 14 01:45:57 2018 daemon.notice procd: /etc/rc.d/S99openwisp config: openwisp: OpenWISP config agent started
Wed Feb 14 01:45:59 2018 daemon.info openwisp: Local configuration outdated
Wed Feb 14 01:45:59 2018 daemon.info openwisp: Downloading configuration from controller...
Wed Feb 14 01:46:00 2018 daemon.info openwisp: Configuration downloaded, now applying it...
Wed Feb 14 01:46:01 2018 daemon.info openwisp: Service network has been reloaded via procd/ubus
Wed Feb 14 01:46:01 2018 daemon.info openwisp: Service system has been reloaded via procd/ubus
Wed Feb 14 01:46:06 2018 daemon.info openwisp: Testing configuration...
Wed Feb 14 01:46:06 2018 daemon.info openwisp: Configuration test succeeded
Wed Feb 14 01:46:06 2018 daemon.info openwisp: Configuration applied successfully
Wed Feb 14 01:52:22 2018 daemon.err openwisp: Failed to connect to controller while getting checksum: curl exit code 28
Wed Feb 14 01:54:22 2018 daemon.info openwisp: Local configuration outdated
Wed Feb 14 01:54:22 2018 daemon.info openwisp: Downloading configuration from controller...
Wed Feb 14 01:54:22 2018 daemon.info openwisp: Configuration downloaded, now applying it...
Wed Feb 14 01:54:23 2018 daemon.info openwisp: Service network has been reloaded via procd/ubus
Wed Feb 14 01:54:23 2018 daemon.info openwisp: Service system has been reloaded via procd/ubus
Wed Feb 14 01:54:28 2018 daemon.info openwisp: Testing configuration...
```

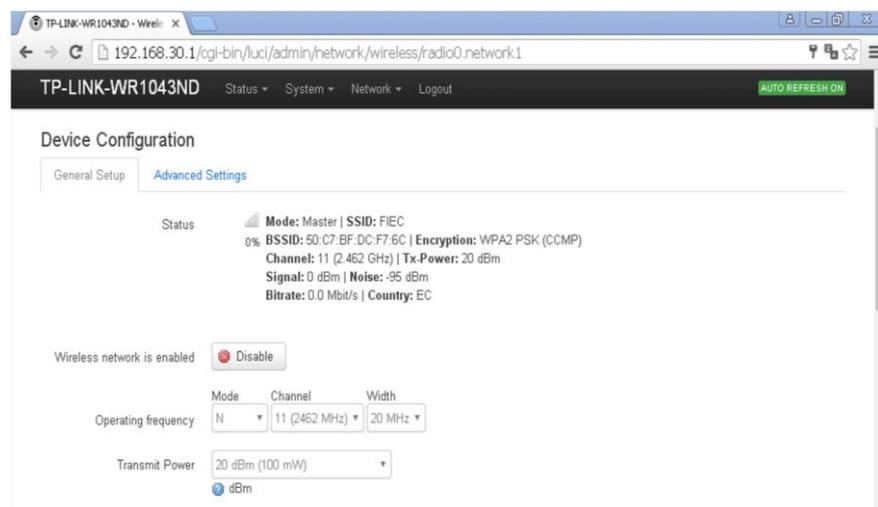
**Figura 4.19** Logs del dispositivo TP-LINK TL-WR1043ND.

Verificada que la configuración de la plantilla se haya cargado completamente en el router inalámbrico se procede a ingresar al router mediante la página web de administración para observar si efectivamente se ha creado la red inalámbrica que fue configurada mediante la plantilla, para esto nos dirigimos a la opción Network y dar clic en Wireless.



**Figura 4.20 Wireless Overview TP-LINK TL-WR1043ND.**

Como se observa en la Figura 4.20 la plantilla WIFI-TPLINK ha creado la red inalámbrica llamada FIEC dentro del dispositivo, para ver más opciones dirigirse a la pestaña Edit donde se tendrán más parámetros como el tipo de canal o el tipo de seguridad que fue definido previamente en la plantilla y que ahora la red inalámbrica del router cuenta con esos parámetros.



**Figura 4.21 Device Configuration TP-LINK TL-WR1043ND.**

La plantilla WIFI-TPLINK puede ser aplicada al dispositivo Cisco Linksys E4200 pero a modo de prueba se va a crear una plantilla diferente con el nombre de WIFI-CISCO con los mismos valores de WIFI-TPLINK pero en la pestaña de canal en la sección Radios se va a colocar un tipo de canal diferente, el canal que se escogió en la plantilla anterior fue 11 y para esta plantilla se va a escoger el canal 6, evitando que los canales no se traslapen, no olvidar colocar el mismo SSID en esta plantilla ya que uno de los objetivos de este proyecto es que todos los router tengan el mismo SSID.

Radio 1	802.11n (2.4 GHz N) ▼
name	radio0
protocol	802.11n ▼
driver	mac80211 ▼
channel	6 ▼
channel width (mhz)	20 ▼
transmit power (dbm)	20
country	Ecuador ▼

**Figura 4.22 Configuración de Radios plantilla WIFI-CISCO.**

Creada la plantilla WIFI-CISCO dirigirse al apartado Devices en el menú principal de Openwisp2 Controller y escoger el dispositivo CISCOLINKSYSE4200 al cual se le va aplicar la plantilla WIFI-CISCO previamente creada, en la parte DEVICE CONFIGURATION DETAILS en la pestaña de Templates se encuentran las plantillas creadas y se selecciona WIFI-CISCO para aplicarla al dispositivo, dar clic en guardar.

Configuration: CISCO-LINKSYSE4200

Backend: OpenWRT/LEDE ▼  
Select [netjsonconfig](#) backend

Status: running  
modified means the configuration is not applied yet; running means applied and running; error means the configuration caused issues and it was rolledback

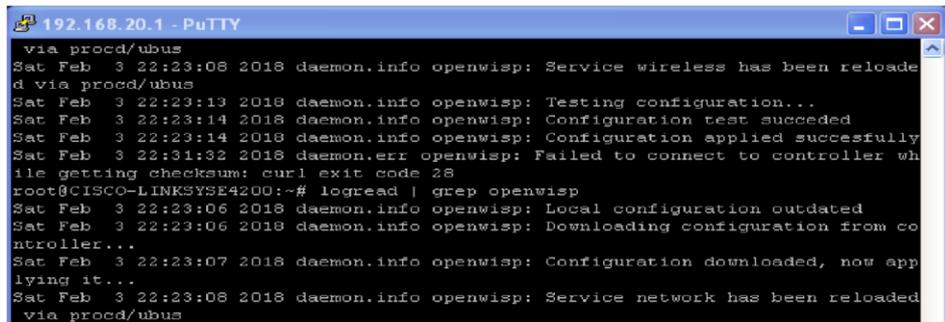
Last ip: 192.168.10.2  
indicates the last ip from which the configuration was downloaded from (except downloads from this page)

Templates:  +

- Zona Horaria
- WIFI-TPLINK
- NTP
- WIFI-CISCO

**Figura 4.23 Uso de la plantilla WIFI-CISCO.**

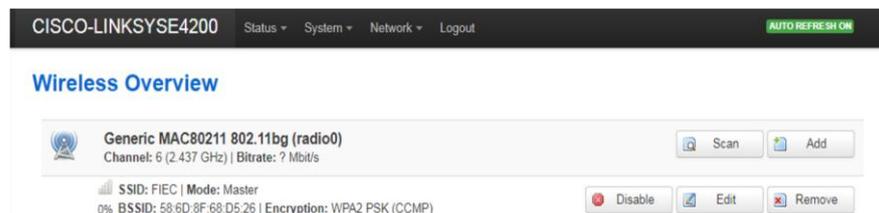
Para verificar si la plantilla se ha aplicado en el dispositivo es necesario ver los logs del dispositivo con el comando `logread | grep openwisp`.



```
192.168.20.1 - PuTTY
via procd/ubus
Sat Feb 3 22:23:08 2018 daemon.info openwisp: Service wireless has been reloaded
d via procd/ubus
Sat Feb 3 22:23:13 2018 daemon.info openwisp: Testing configuration...
Sat Feb 3 22:23:14 2018 daemon.info openwisp: Configuration test succeeded
Sat Feb 3 22:23:14 2018 daemon.info openwisp: Configuration applied successfully
Sat Feb 3 22:31:32 2018 daemon.err openwisp: Failed to connect to controller wh
file getting checksum: curl exit code 28
root@CISCO-LINKSYS-E4200:~# logread | grep openwisp
Sat Feb 3 22:23:06 2018 daemon.info openwisp: Local configuration outdated
Sat Feb 3 22:23:06 2018 daemon.info openwisp: Downloading configuration from co
ntroller...
Sat Feb 3 22:23:07 2018 daemon.info openwisp: Configuration downloaded, now app
lying it...
Sat Feb 3 22:23:08 2018 daemon.info openwisp: Service network has been reloaded
via procd/ubus
```

**Figura 4.24** Logs del dispositivo CISCO-LINKSYS-E4200.

Verificada que la configuración de la plantilla WIFI-CISCO se haya cargado completamente en el router inalámbrico se procede a ingresar al router mediante la página web de administración para observar si efectivamente se ha creado la red inalámbrica que fue creada mediante la plantilla, para eso dirigirse a la opción Network y dar clic en Wireless.



**Figura 4.25** Wireless Overview CISCO-LINKSYS-E4200.

### 4.2.3 Pruebas de la red inalámbrica FIEC.

Una vez creadas las plantillas y aplicarlas en ambos routers se procede a probar la red inalámbrica FIEC que fueron creadas en ambos dispositivos, para esto se encienden ambos routers inalámbricos y desde un dispositivo portátil sea una laptop o un celular escaneamos las redes WIFI y buscamos la red con el nombre FIEC.



**Figura 4.26 Red inalámbrica FIEC.**



**Figura 4.27 Ingreso a la Red FIEC.**



**Figura 4.28 Conexión a la Red FIEC.**

Como se observa en la Figura 4.26 solo se visualiza una red con el nombre FIEC esto se debe a que ambos routers fueron configurados con el mismo SSID ya que ambas plantillas poseen la misma configuración solo se diferencian en el canal de transmisión es por eso que el usuario que se quiera conectar a cualquiera de los dos dispositivos solo va a visualizar una sola red con el nombre FIEC que será la red que emita una mejor señal en ese instante.

Por ejemplo un usuario se conecta al router TP-LINK coloca la clave la contraseña y está navegando en esa red, la ip que le asigna el router en ese momento es la 192.168.30.103.



**Figura 4.29 Red FIEC router TP-LINK.**

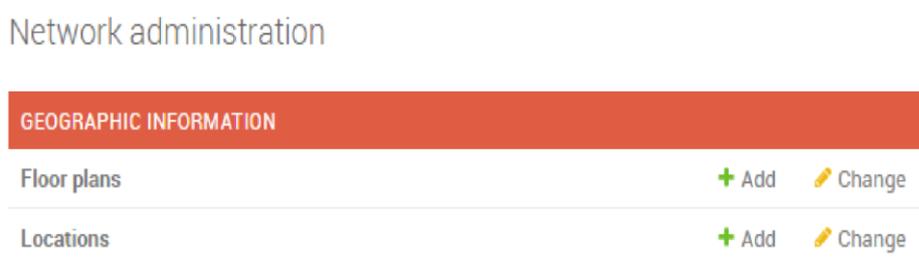
Luego el usuario se mueve a otro sitio y pierde señal con el router TP-LINK pero en ese sitio está el otro router CISCO en este caso el usuario se conectara automáticamente a ese router el cual le asigna la dirección ip 192.168.20.220 siendo este cambio transparente para el usuario.



**Figura 4.30 Red FIEC router CISCO.**

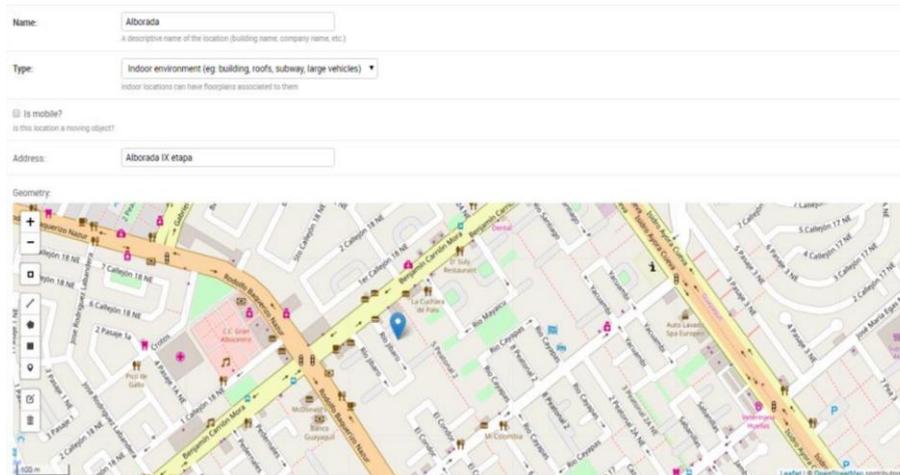
### 4.3 Información Geográfica.

En el menú principal de Openwisp2 Controller se pueden crear localizaciones para el controlar la ubicación de los dispositivos, para crearlas dar clic en Locations.



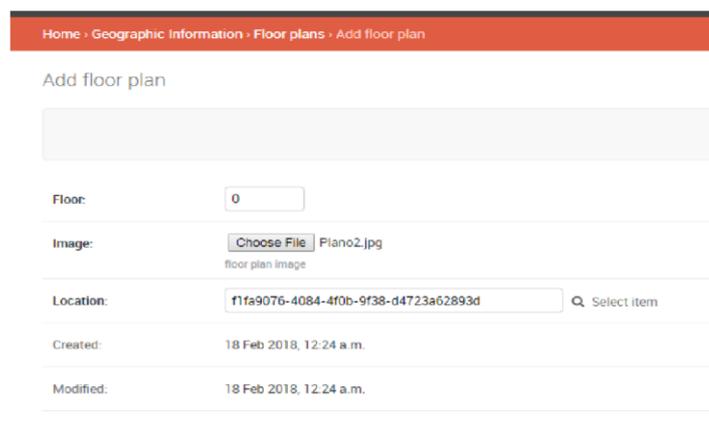
**Figura 4.31 GEOGRAPHIC ADMINISTRATION.**

Elegir el nombre de la localización, en el tipo de lugar se escogió interior para ejemplo debido a que los routers están dentro de un domicilio, en Address llenar el campo con la dirección domiciliaria y dar clic en guardar.



**Figura 4.32 Creación de mapa de los routers.**

Para complementar una localización se pueden crear Floor plans en donde se carga un mapa previamente creado del interior del domicilio y de esta manera ver la ubicación exacta de cada dispositivo, es de gran utilidad, ya que se pueden crear mapas de infraestructuras de red con varios dispositivos y así llevar un control absoluto de la ubicación de cada dispositivo en la red.



**Figura 4.33 Creación de un Floor plans.**

Para asociar una ubicación geográfica a los dispositivos se ingresa en cada router y en la sección de GEOGRAPHIC INFORMATION elegir la localización creada.

GEOGRAPHIC INFORMATION

Geographic information: #1

Location selection: --- Please select an option ---

**Figura 4.34 Location selection.**

Por último, ir a la pestaña INDOOR COORDINATES y elegir el Floor plans creado, clic sobre la imagen del dispositivo al que pertenece y luego guardar, automáticamente la ubicación geográfica se asocia al dispositivo.

INDOOR COORDINATES

Floorplan selection: Existing

Floorplan: Alborada ground floor

Floor: 0

Image: Choose File No file chosen  
floor plan image

indoor position

+  
-

**Figura 4.35 Indoor Coordinates.**

## CONCLUSIONES Y RECOMENDACIONES

Se logró implementar un software controlador de código abierto en la distribución Ubuntu server para la administración de routers inalámbricos.

Con el uso de Openwrt como firmware base de los routers inalámbricos se obtuvo libertad en la configuración de los dispositivos mediante la interfaz de línea de comandos.

Con la ayuda del paquete openwisp-config-openssl se consiguió registrar los dispositivos a Openwisp2 Controller.

Se desarrolló plantillas para configuración de la zona horaria de los dispositivos.

Se creó plantillas para la configuración de una red inalámbrica para cada dispositivo registrado al controlador.

Se logró configurar un único SSID para los dispositivos registrados al controlador.

Con la creación de Floor plans se consiguió que los routers inalámbricos tengan información de su ubicación geográfica mediante mapas creados por el usuario.

Al momento de instalar Ubuntu server configurar correctamente las interfaces de red, una interfaz debe ser las que nos provee internet y la otra interfaz deber ser para la red interna.

Verificar que los Routers inalámbricos cuenten con una memoria flash mayor a 8MB para que al momento de instalar paquetes no se den errores de falta de memoria en el dispositivo.

Si la versión del firmware Openwrt para el dispositivo no está disponible en la página oficial se puede trabajar con LEDE que es un firmware basado en Openwrt que funciona perfectamente en los routers inalámbricos.

Trabajar con el paquete openwisp-config-openssl para la configuración del registro del dispositivo debido a que es estable y no da problemas al momento de la instalación.

Al crear las plantillas para una red inalámbrica verificar que el canal de transmisión de datos no sea el mismo, preferible trabajar con los canales 1, 6 y 11.

Revisar logs de los routers inalámbricos al momento de realizar cambios en el controlador y así verificar que se están efectuando correctamente.

Instalar Openwisp2 Controller en la distribución de Ubuntu server, ya que no genera errores al momento de instalar las herramientas, paquetes y complementos adicionales que necesita el controlador.

## BIBLIOGRAFÍA

- [1] OpenWrt, «welcome to the openWrt,» s.f.. [En línea]. Available: <https://openwrt.org/>. [Último acceso: 18 01 2018].
- [2] OpenWISP, «What is OpenWISP?,» s.f.. [En línea]. Available: <http://openwisp.io/docs/general/values.html>. [Último acceso: 10 01 2018].
- [3] OpenWISP, «OpenWM (OpenWISP Manager) and OpenWF (OpenWISP Firmware),» s.f.. [En línea]. Available: <http://openwisp.org/whatis.html>. [Último acceso: 11 01 2018].
- [4] OpenWISP, «OpenWISP Firmware - Description,» s.f.. [En línea]. Available: <https://github.com/openwisp/OpenWISP-Firmware#description>. [Último acceso: 11 01 2018].
- [5] OpenWISP, «OpenWGM (OpenWISP Geographic Monitoring),» s.f.. [En línea]. Available: <http://openwisp.org/whatis.html>. [Último acceso: 11 01 2018].
- [6] Ansible, «Ansible Documentation - about Ansible,» s.f.. [En línea]. Available: <http://docs.ansible.com/ansible/latest/index.html>. [Último acceso: 12 01 2018].
- [7] A. De Andrés Iema, «Características de Ubuntu Server,» 28 02 2010. [En línea]. Available:
- [8] [https://manuais.iessanclemente.net/index.php/Caracter%C3%ADsticas\\_de\\_Ubuntu\\_Server](https://manuais.iessanclemente.net/index.php/Caracter%C3%ADsticas_de_Ubuntu_Server). [Último acceso: 13 01 2018].
- [9] Debian, «Razones para escoger Debian,» s.f.. [En línea]. Available: [https://www.debian.org/intro/why\\_debian.es.html](https://www.debian.org/intro/why_debian.es.html). [Último acceso: 13 01 2018].

[10] c. Ethernet, «¿QUÉ ES UN CABLE ETHERNET?,» Tecnología fácil, 12 07 2015. [En línea]. Available: <https://tecnologia-facil.com/que-es/que-es-un-cable-ethernet/>. [Último acceso: 13 01 2018].

[11] tp-link, «Routers Inalámbricos TL-WR1042ND,» s.f.. [En línea]. Available: <http://www.tplink.com/es/products/details/TL-WR1042ND.html>. [Último acceso: 14 01 2018].

[12] S. D. Luz, «Cisco Linkys E4200,» 28 06 2011. [En línea]. Available: <https://www.redeszone.net/cisco-linksys/e4200/>. [Último acceso: 14 01 2018].

[13] tp-link, «Descripción General,» s.f.. [En línea]. Available: [https://www.tplink.com/pe/products/details/cat-4763\\_TL-SF1008D.html](https://www.tplink.com/pe/products/details/cat-4763_TL-SF1008D.html). [Último acceso: 15 01 2018].

[14] tp-link, «Accesorios Electrónicos - Visión General,» s.f.. [En línea]. Available: [https://www.tplink.com/es/products/details/cat-5688\\_UE300.html](https://www.tplink.com/es/products/details/cat-5688_UE300.html). [Último acceso: 14 01 2018].