



# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

## **Facultad de Ingeniería en Electricidad y Computación**

REDUCIR EL TIEMPO DE SOLUCIÓN ANTE REPORTES DE FALLAS  
DE LAS REDES INALÁMBRICAS EN HACIENDAS DEL SECTOR  
AGRÍCOLA BANANERO PARA LA EMPRESA “PROCESOS IQ”

### **PROYECTO INTEGRADOR**

Previo la obtención del Título de:

**Licenciado en Redes y Sistemas Operativos**

Presentado por:

Sanny Abigail Florencia Guzmán

GUAYAQUIL - ECUADOR

Año: 2018

## **DEDICATORIA**

El presente proyecto lo dedico a Félix Guzmán y Gladys Chalén por todo el amor, apoyo y cariño que me dieron.

## **AGRADECIMIENTOS**

Mi más sincero agradecimiento a mis profesores, compañeros y las personas que estuvieron involucradas en el desarrollo de este proyecto.

## **DECLARACIÓN EXPRESA**

“Los derechos de titularidad y explotación, me corresponde conforme al reglamento de propiedad intelectual de la institución; *Sanny Abigail Florencia Guzmán* y doy mi 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”

Sanny Abigail Florencia Guzmán

## **EVALUADORES**

**Ing. Robert Stalin Andrade Troya**

PROFESOR DE LA MATERIA

**Ing. Jorge Antonio Magallanes Borbor**

PROFESOR TUTOR

## RESUMEN

La empresa "Procesos IQ" con sede en Guayaquil tiene como modelo de negocio brindar servicios tecnológicos a productores agrícolas que se encuentran ubicados en sectores cercanos a Quevedo.

Uno de los servicios que ofrece es la instalación, administración y soporte técnico de redes inalámbricas. El servicio de soporte técnico que ofrece la empresa siempre es presencial, esto quiere decir que no importa que se requiera un técnico en el sector o no, siempre un técnico viajará a hacia Quevedo, debido a esto y por encontrarse tan distante de sus clientes, para la empresa realizar viajes constantemente puede representar un gasto de tiempo y dinero en aquellos casos que la solución puede darse remotamente.

El objetivo de este proyecto es encontrar una solución que permita a la empresa para aumentar la rentabilidad, evitando realizar viajes innecesarios y ahorrando tiempo a la empresa y sus empleados.

Con la metodología del Design Thinking, se permitió conocer a fondo al cliente, encontrar sus puntos débiles, y encontrar una solución que se adapte a sus necesidades. Esta es una solución que usa software libre, para el sitio web usando Python + Django, y la base de datos usando PostgreSQL, estará alojada en la nube por lo que podrá ser accedida en cualquier lugar, teniendo los permisos necesarios.

**Palabras Clave:** Design Thinking, servicio técnico, software libre, redes inalámbricas.

## **ABSTRACT**

*The company "Procesos IQ" based in Guayaquil has as a business model to provide technological services to agricultural producers that are located in sectors near Quevedo.*

*One of the services offered is the installation, administration and technical support of wireless networks. The technical support service offered by the company is always face-to-face, this means that it does not matter if a technician is required in the sector or not, always a technician will travel to Quevedo, because of this and because he is so distant from his clients, for the company to make trips constantly can represent an expense of time and money in those cases that the solution can be given remotely.*

*The objective of this project is to find a solution that allows the company to increase profitability, avoiding unnecessary trips and saving time for the company and its employees.*

*With the methodology of Design Thinking, it allowed to know the customer thoroughly, find their weak points, and find a solution that suits their needs. This is a solution that uses free software, for the website using Python + Django, and the database using PostgreSQL, will be hosted in the cloud so it can be accessed anywhere, having the necessary permissions.*

*Keywords: Design Thinking, technical service, free software, wireless networks.*

# ÍNDICE GENERAL

RESUMEN.....	I
<i>ABSTRACT</i> .....	II
ÍNDICE GENERAL.....	III
ABREVIATURAS .....	V
SIMBOLOGÍA .....	VI
ÍNDICE DE FIGURAS.....	VII
ÍNDICE DE TABLAS .....	VIII
CAPÍTULO 1 .....	1
1. INTRODUCCIÓN .....	1
1.1 Descripción del problema .....	2
1.2 Justificación del problema.....	2
1.3 Objetivos.....	3
1.3.1 Objetivo General .....	3
1.3.2 Objetivos Específicos .....	3
CAPÍTULO 2.....	4
2. METODOLOGÍA .....	4
CAPÍTULO 3.....	11
3. RESULTADOS Y ANÁLISIS .....	11
3.1.1 Funcionalidades .....	14
3.1.1 Configuración de datos .....	14
3.1.2 Perfiles de usuario.....	15
3.1.3 Ubicación geográfica de antenas .....	16
3.1.4 Monitoreo en tiempo real.....	18



3.1.5	Notificación de alertas .....	20
3.1.6	Gestión de errores, reglas, incidencias .....	20
3.1.7	Reportes.....	22
3.1.8	Sistema de alta disponibilidad y balanceo de carga.....	23
3.1.9	Base de datos tolerante a fallas .....	24
CAPÍTULO 4.....		25
4	IMPLEMENTACIÓN Y PRESUPUESTO .....	25
4.1	Plan de implementación .....	<b>¡Error! Marcador no definido.</b>
4.2	Presupuesto.....	25
CONCLUSIONES Y RECOMENDACIONES .....		26
Conclusiones .....		26
Recomendaciones .....		26
BIBLIOGRAFÍA.....		28
ANEXOS.....		29

## ABREVIATURAS

API	Application Programming Interface
CPU	Central Process Unit
CSS	Cascading Style Sheets
ESPOL	Escuela Superior Politécnica del Litoral
GPS	Global Positioning System
HTML	HyperText Markup Language
ICMP	Internet Control Message Protocol
IP	Internet Protocol
ISP	Internet Service Provider
IT	Information Technology
JS	JavaScript
MAC	Media Access Control
MIMO	Multiple Input Multiple Output
MTU	Maximum Transmission Unit
PING	Packet Internet Group
PySNMP	Python Simple Network Management Protocol
RAM	Random Access Memory
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
SSD	Solid State Drive
SSH	Secure Shell
URL	Uniform Resource Locator

## **SIMBOLOGÍA**

DBi	Decibelios
GB	Gigabyte
Km	Kilómetros
TB	Terabyte

## ÍNDICE DE FIGURAS

Figura 2.1 Mapa stakeholders internos y externos de la empresa.....	4
Figura 2.2 Puntos de vista usuarios de la empresa .....	5
Figura 2.3 Herramienta ¿Cómo podríamos? .....	6
Figura 2.4 Diagrama causa y efecto .....	7
Figura 2.5 Datos a ingresar por cliente .....	8
Figura 2.6 Mapa de antenas que tienen los diferentes clientes .....	9
Figura 2.7 Proceso de prototipar y testear.....	10
Figura 3.1 Arquitectura de la aplicación.....	12
Figura 3.2 Actividades que realizará el usuario .....	14
Figura 3.3 Registro de cliente en el sistema .....	15
Figura 3.4 Registro de equipos .....	16
Figura 3.5 Registros de antenas del cliente .....	17
Figura 3.6 Mapa de antenas registradas .....	17
Figura 3.7 Listado de escaneos realizados.....	18
Figura 3.8 Niveles de acceso hacia dispositivos de red.....	18
Figura 3.9 Proceso de notificación de errores y asignación de incidencia.....	21
Figura 3.10 Ventana de registro de incidencias.....	22
Figura 3.11 Reporte de incidencias .....	23
Figura 3.12 Arquitectura de alta disponibilidad y balanceo de carga.....	23
Figura 3.13 Base de datos Failover .....	24
Figura 4.1 Plan de implementación.....	25

## ÍNDICE DE TABLAS

Tabla 2.1 Matriz decisión .....	7
Tabla 3.1 Módulos que serán instalados en la aplicación.....	13
Tabla 4.1 Presupuesto implementación.....	25



# CAPÍTULO 1

## 1. INTRODUCCIÓN

La empresa con sede en Guayaquil se dedica al desarrollo e implementación de aplicaciones web y móviles orientadas a la agricultura, tales como monitoreo en tiempo real de procesos de producción, control de asistencia y planificación de trabajos realizados por jornaleros, asistencia para la prevención de Sigatoka, control de parámetros de calidad de los procesos de producción, monitoreo de las condiciones climáticas en tiempo real, bitácora de rechazos y sus causas, además de aplicaciones de control de inventario y control de órdenes de trabajo usadas en otras industrias ajenas a la agricultura, también el diseño e implementación de infraestructuras de redes inalámbricas.

Las aplicaciones están conformadas por la captura de datos en tiempo real, ya sea por algún usuario final a través de la aplicación móvil, o mediante la captura de datos automática de dispositivos electrónicos. Estos datos capturados sirven para alimentar el sistema principal, que es una web alojada en una nube privada Digital Ocean que permite a supervisores, monitorear y corregir la operación, y a gerentes a visualizar reportes.

Al momento cuentan con 25 clientes, que incluyen el agro y otras industrias, algunos de estos clientes contrataron los servicios de implementación de antenas de radio, 8 de estos clientes actualmente reciben soporte de red.

Un caso de un cliente que usa todas las aplicaciones cuenta con dos haciendas y cuenta con personal administrativo que en su mayoría opera desde la ciudad de Guayaquil y personal agrícola que está en cada hacienda y constan de un jefe y un asistente de RRHH, una enfermera, dos cocineras, dos guardias de seguridad, un jefe de bodega, dos bodegueros, cuatro supervisores de calidad, ocho supervisores de trabajos agrícolas y alrededor de doscientos cincuenta trabajadores agrícolas.

## **1.1 Descripción del problema**

Los clientes necesitan conexión a Internet para acceder a las aplicaciones que ofrece la empresa, y también para correos, y mensajería instantánea.

Antes el cliente contrataba con un ISP el servicio de internet mediante un enlace dedicado en cada hacienda, con el tiempo el cliente creció, adquirió más haciendas y contratar el servicio de internet en cada una de ellas le resultó más costoso, debido a esto la empresa, empezó a brindar el servicio de implementación de infraestructuras de redes inalámbricas con alquiler, lo cual permitió a sus clientes contratar un solo enlace y a partir de él crear su propia red para así conectar el resto de sus haciendas.

Debido a la buena acogida de esta implementación, cada vez tienen más clientes que requieren del servicio, que, si bien es un beneficio para la empresa, provocó la necesidad de horas/hombre para el mantenimiento de los equipos y redes instalados.

El tiempo que se toma atender problemas de red presentados en las haciendas, puede variar de 6 horas a 2 días dependiendo de la gravedad, esto provoca molestias a la empresa debido a los costos de transporte y viáticos, la baja rentabilidad, y el tiempo perdido al realizar el viaje a Quevedo desde Guayaquil.

## **1.2 Justificación del problema**

Existen problemas de red sencillos que se pueden atender y/o solucionar con gestión remota y no necesita técnicos en las haciendas por el contrario hay problemas más complejos los cuales sí necesitan que un técnico viaje para dar asistencia local.

Los costos que representan mantener personal de soporte cerca de las haciendas de los clientes, realizar constantemente viajes de Guayaquil a Quevedo y viceversa, pueden resultar muy costosos.



Minimizando la cantidad de viajes que el técnico realiza, la empresa podrá reducir sus costos, lo que generará mayor rentabilidad, así como también se disminuirá el tiempo de respuesta de la solución del problema generando mayor satisfacción al usuario.

### **1.3 Objetivos**

#### **1.3.1 Objetivo General**

Disminuir costos de operación en el proceso de soporte técnico de redes, al diagnosticar el tipo de atención requerida, mediante monitoreos de equipos y gestión de incidencias, para aumentar la rentabilidad de la empresa.

#### **1.3.2 Objetivos Específicos**

1. Diseñar una solución informática que permita reducir cantidad de viajes innecesarios hacia Quevedo de manera que se genere mayor rentabilidad.
2. Crear un prototipo que permita monitorear equipos de red y generar incidencias mediante la ejecución de comandos de red para notificar a los técnicos.
3. Diseñar reportes que se incluyen en el prototipo con el fin de manejar fallas de comunicación en las redes y evaluar tiempos de detección de problemas y de respuestas.
4. Asociar errores generados en las incidencias con posibles soluciones para poder saber cómo se solucionó el problema presentado.

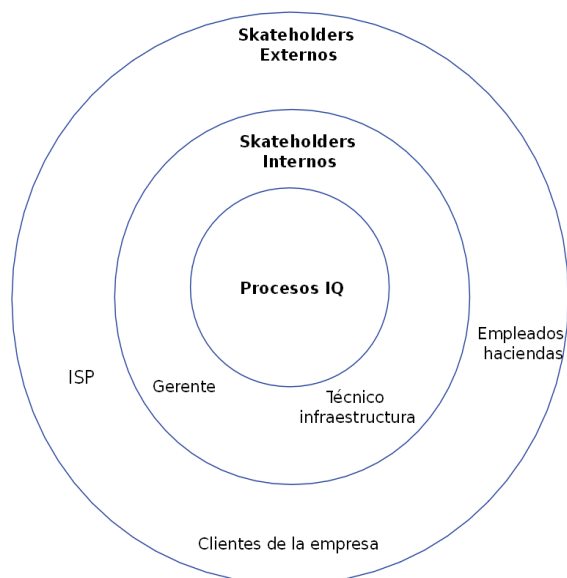
# CAPÍTULO 2

## 2. METODOLOGÍA

En este capítulo se describen los resultados obtenidos a través de la metodología “Design Thinking” que se ha empleado en la empresa “Procesos IQ”.

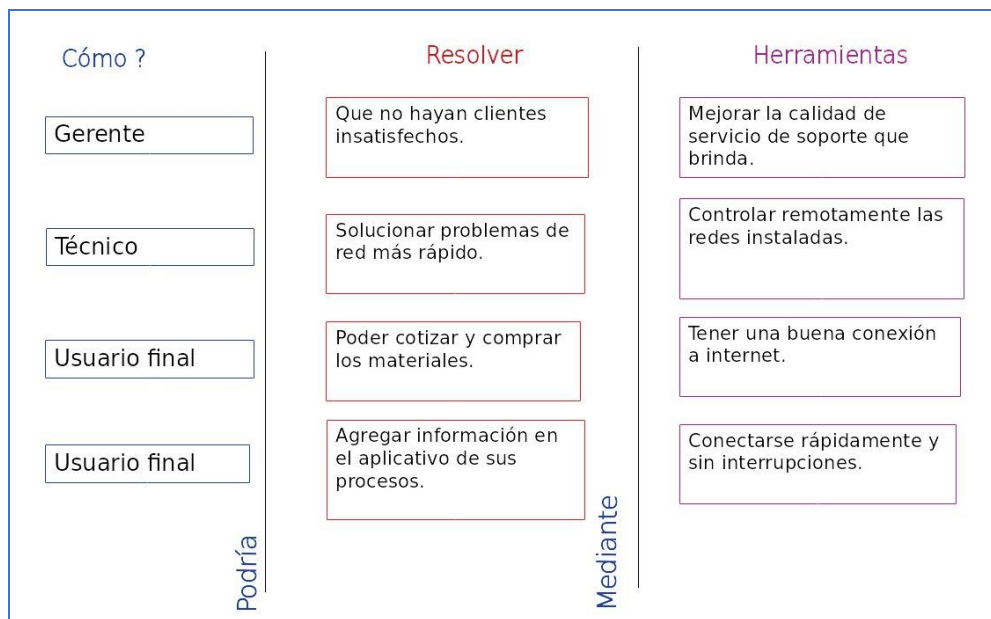
Se empezó realizando una investigación de escritorio [Anexo A] que permitió conocer a la empresa, saber cuáles son las actividades que realiza, también cuál es su modelo de negocios y quiénes interactúan con ella, para poder tener un mayor acercamiento con nuestro cliente, saber sus preferencias, necesidades y sus molestias.

Definimos un diagrama denominado stakeholders el cual muestra cuales son los actores ya sean empleados, proveedores, clientes que están involucrados interna o externamente con la empresa. En la figura 2.1 tenemos como stakeholders externos a los clientes de la empresa, los empleados de las haciendas de los clientes y al ISP que provee del enlace dedicado a cada cliente. Como stakeholders internos tenemos al gerente de la empresa, empleados técnicos de infraestructura.



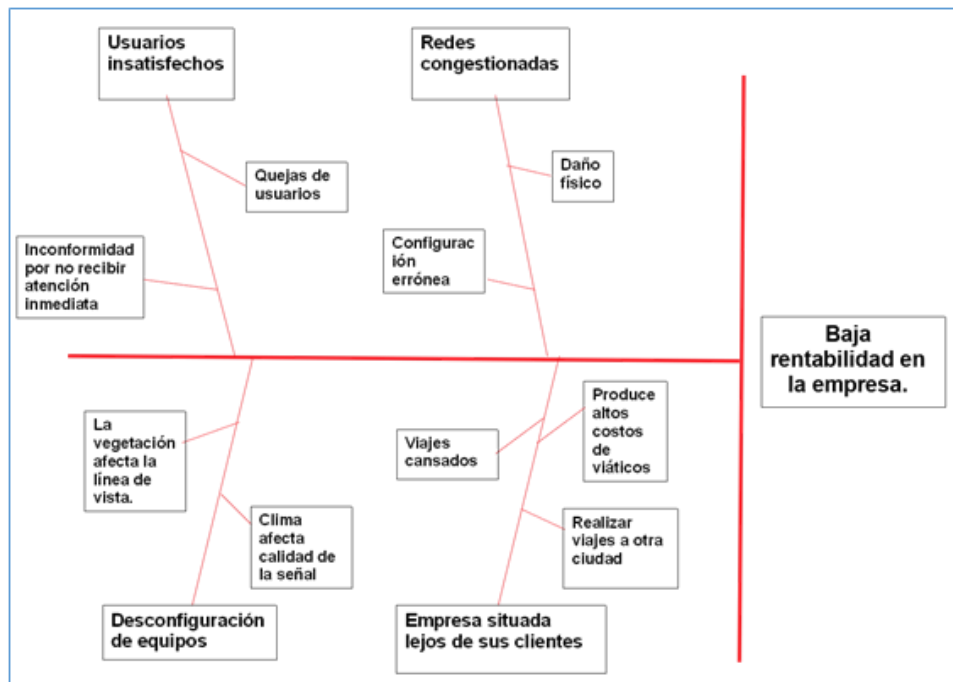
**Figura 2.1 Mapa stakeholders internos y externos de la empresa**





**Figura 2.3 Herramienta ¿Cómo podríamos?**

Con los resultados obtenidos en todos los procesos anteriores, se generó un diagrama figura 2.4 denominado causa-efecto o espina de pescado. Se compone de una parte principal o sea la cabeza de pescado donde definiremos la problemática global, luego de eso se muestran en las espinas las causas de la problemática y los efectos que provoca cada una.



**Figura 2.4 Diagrama causa y efecto**

Yo pienso que soluciones puedo ofrecer para resolver los problemas. La tabla 2.1 muestra nuestra matriz decisión que evalúa todas las soluciones en base a los problemas y necesidades que tiene la empresa. La calificación puede ser de 0 a 3, donde 0 no satisface y 3 es completamente satisfactorio.

**Tabla 2.1 Matriz decisión**

Alternativas solución / Problemas necesidades	Colocar una sucursal en la ciudad de Quevedo.	Diseño de un sistema de monitoreo de redes y gestión de problemas.	Contratar más personal técnico para supervisor.
Baja rentabilidad	1	2	1
Poco personal	0	3	2
Alejados de los clientes	3	2	0
Mucho tiempo de viaje	3	1	0

<b>Poco control de los servicios técnicos ejecutados</b>	0	3	3
<b>Usuarios insatisfechos</b>	1	2	1
<b>Total</b>	<b>8</b>	<b>11</b>	<b>7</b>

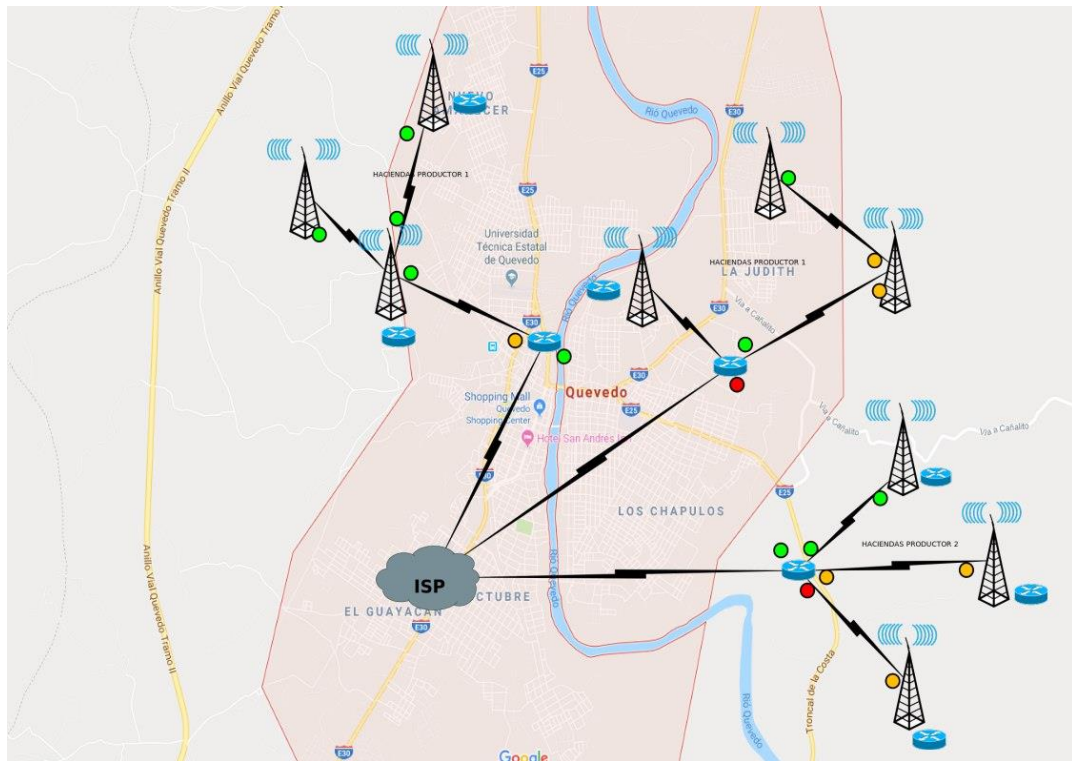
Se ha considerado que tener una sucursal cerca de Quevedo, no aportaría de forma significativa la reducción de costos, porque de todas formas los problemas de red continuarían y no habría una buena gestión de soporte.

Por la solución planteada como diseño de un sistema de monitoreo puede ser la mejor opción para el monitoreo, detección y prevención de cualquier anomalía en la red, ya que gracias a esta solución podremos recibir notificaciones de que problemas se han presentado, para así poder decidir si el técnico tiene que viajar o no, esto sería de gran ayuda en el ahorro de tiempo y viajes innecesarios.

Cliente	Sanny Florencia
Equipo	Cisco
IP	192.168.1.12
Coordenadas	-1.0954891,-79.6240031

**Figura 2.5 Datos a ingresar por cliente**

Se realizó un prototipo de la solución para mostrar al cliente, el cual fue elegido en base a los criterios de la matriz de decisión. En la figura 2.5 se observa una pantalla para el ingreso de datos del cliente que el usuario técnico tendrá que ingresar, porque una vez ingresada la información se va a poder observar en la figura 2.6 un mapa donde están ubicadas todas las antenas y equipos provenientes del cliente de la empresa. Cada dispositivo cuenta con un botón circular que puede tener tres tipos de colores según su estado: verde si está disponible, naranja si el enlace esta intermitente, y rojo si no hay acceso.

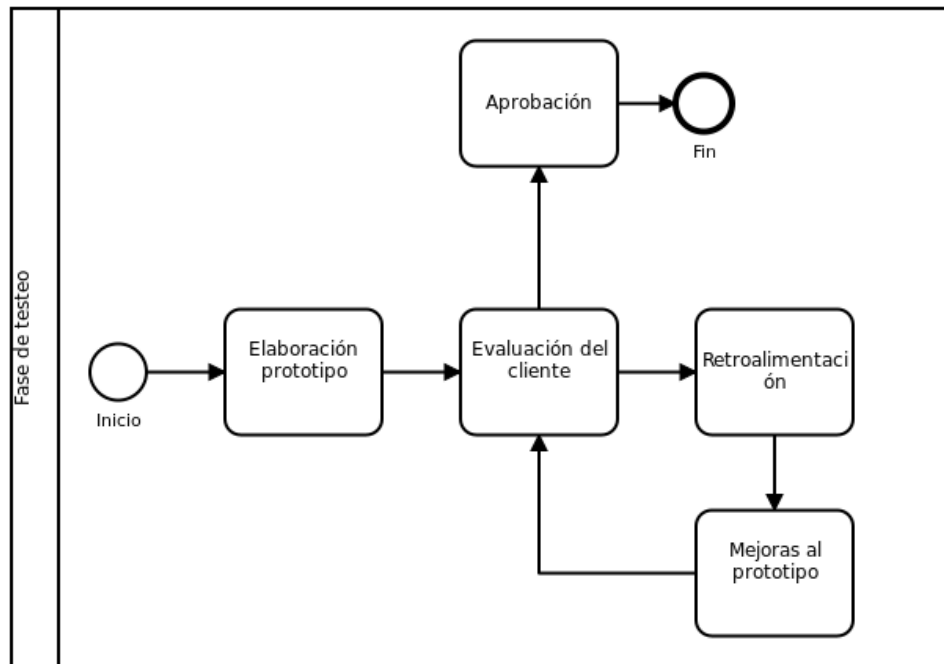


**Figura 2.6 Mapa de antenas que tienen los diferentes clientes**

Cuando mostramos el prototipo al cliente este hizo sus observaciones y sus requerimientos los cuales tomamos en cuenta para poder mejorar el prototipo, entre los requerimientos del cliente podemos listar los siguientes:

- Compatible con todas las marcas que usa la empresa
- Notifique al usuario cuando se presente una falla
- Poder acceder remotamente
- Generación de reportes

Este proceso de revisiones con el cliente y en clases lo podemos describir con la siguiente figura 2.7 donde el primer paso es la elaboración del prototipo luego son las reuniones con el cliente y en caso de que exista retroalimentación se realizan las mejoras y se vuelve al ciclo hasta que el prototipo sea aprobado.



**Figura 2.7** Proceso de prototipar y testear

En el siguiente capítulo estará la solución elegida , con los requerimientos y observaciones hechas por el cliente y el profesor de la materia integradora.



# CAPÍTULO 3

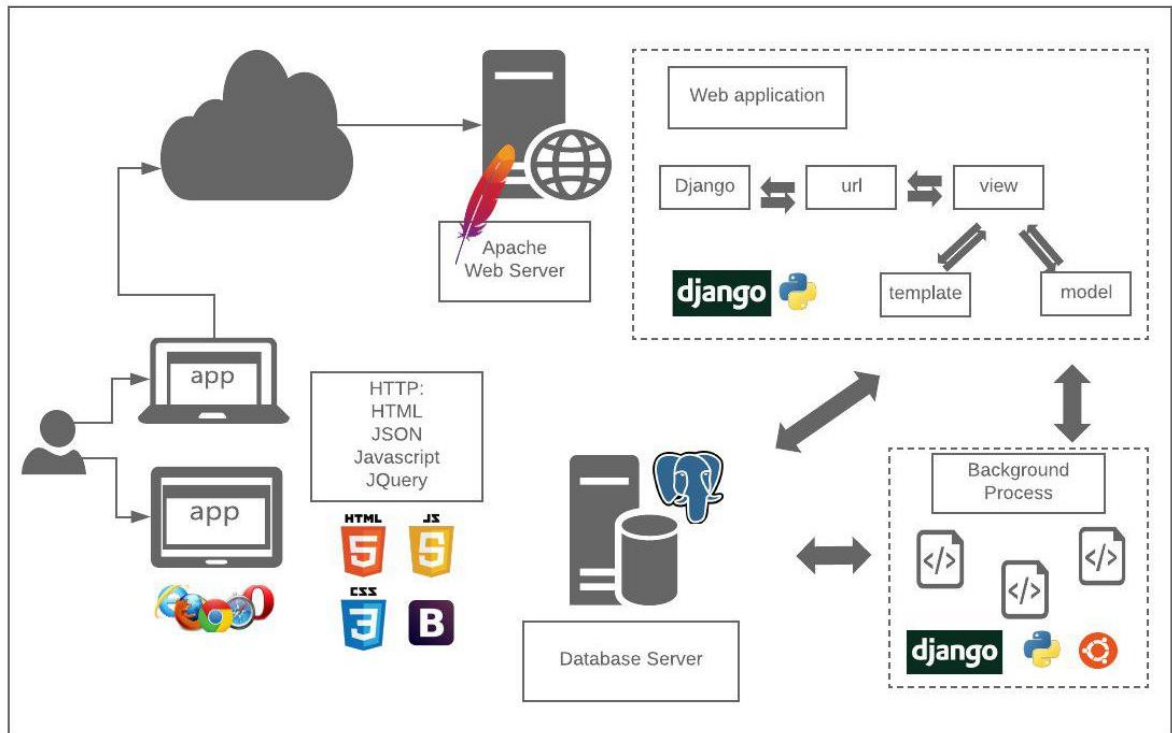
## 3. RESULTADOS Y ANÁLISIS

El sistema por diseñar constituye una plataforma web [Anexo C], cuya función principal será la ejecución de comandos para escanear los dispositivos de red que se encuentran instalados en las diferentes haciendas de los clientes. Contará con una sección administrativa que permitirá añadir estos dispositivos de red para su posterior escaneo. Así mismo, se podrá configurar otros aspectos tales como clientes, usuarios y perfiles. Además, cada dispositivo tendrá una bitácora de eventos, y se generarán alertas cuando estos eventos presenten alguna anomalía.

Esta plataforma estará diseñada para empresas que se encarguen de la venta radio enlaces y accesos a internet y también realicen la gestión de estos. Por esto, para hacer uso de la plataforma es importante que el usuario tenga su topología de red definida, la información de dirección IP, usuarios administrativos y contraseñas, además coordenadas geográficas de los equipos. Además, el equipo necesita de ser el caso tener acceso de las redes, esto va a depender única y exclusivamente de la autorización de los clientes.

La figura 3.1 muestra la arquitectura de toda la aplicación, el usuario podrá realizar peticiones al servidor web alojado en la nube, con una Tablet o una computadora. Este servidor web, en Apache [6], asignará las tareas solicitadas por el usuario a la aplicación hecha en Django Web Framework.

El recuadro de Background process de la figura 3.1, representa al conjunto de scripts que serán ejecutados como aplicaciones en segundo plano para monitorear los equipos de red, que consultarán y crearán nuevos registros en la base de datos.



**Figura 3.1 Arquitectura de la aplicación**

Se usará como sistema operativo, Ubuntu LTS 18.04. Entre las configuraciones básicas que el servidor debe tener se muestran las siguientes:

- Creación de usuarios
- Instalación de módulos descritos en la Tabla
- Activar el servicio SSH
- Eliminar el acceso por SSH del usuario root
- Eliminar el acceso por SSH a través de contraseñas
- Cambiar de puerto por defecto de SSH
- Generar una clave SSH en el servidor
- Compartir la clave pública generada con otros equipos
- Instalación servicio HTTP

Además, el sistema cuenta con los siguientes módulos mostrados en la tabla 3.1

**Tabla 3.1 Módulos que serán instalados en la aplicación**

Herramientas	Versión	Descripción
<b>Django</b>	2.0	Framework para creación de sitios web.
<b>Python</b>	3.7	Lenguaje de programación
<b>PostgreSQL</b>	10.4	Base de datos
<b>Javascript</b>		Lenguaje de programación
<b>Modulos adicionales</b>		
<b>django-jet</b>	1.7.0	Personalización de las plantillas de administrador de Django.
<b>django-map-widgets</b>	0.1.9	Personalización de campos geográficos en modelo de datos
<b>psycopg2</b>	2.7.5	Adaptador de Postgresql para el lenguaje Python
<b>Subprocess[5]</b>		Ejecución de procesos de Linux desde Python
<b>Custom management commands[4]</b>		Creación de scripts que se ejecutan con el admin de Django.
<b>Jquery</b>	3.3.1	Librería de JavaScript que permite manipular documentos HTML.
<b>Bootstrap</b>	4.1.3	Librería para el desarrollo de HTML, CSS y JS
<b>Google Maps API[2]</b>	3.34	Plataforma que permite trabajar con mapas en una página web.
<b>Paramiko</b>	2.4.1	Permite realizar conexiones via SSH
<b>PySNMP</b>	4.4.5	Herramienta que permite la comunicación con dispositivos sin importar marcas, usando el protocolo SNMP.

### 3.1.1 Funcionalidades

En el sistema el usuario de soporte técnico podrá realizar las actividades descritas en la figura 3.2. A continuación, el detalle de cada una de las funcionalidades de la solución:



**Figura 3.2 Actividades que realizará el usuario**

#### 3.1.1. Configuración de datos

Un usuario podrá registrar, editar y eliminar información de clientes y sus dispositivos, ubicándolos en un mapa para saber su posición correcta, además podrá registrar nuevos usuarios y nuevos perfiles con diferentes tipos de permisos para acceder a la aplicación.

Es necesario que el usuario realice un ingreso de información inicial, para saber cuáles son los equipos que se van a escanear, a quien le pertenecen, etc., como funcionalidad adicional el usuario podrá consultar todos los datos registrados y los generados por escaneos, también podrá generar reportes de estos datos.

Tenemos como ejemplo en la figura 3.3 la creación de un nuevo cliente, el cual consta de información como nombre, correo electrónico y ubicación en el mapa. En la siguiente pestaña la opción de añadir tantas antenas como posea el cliente.

GENERAL ANTENAS

Nombre:\* Cliente Prueba

Correo electronico:\* sbflorence@gmail.com

Coordenadas: Coordinates Use My Location Enter or update the address here

Map Satellite

Quevedo

Valencia

La Maná

Velasco Ibarra

LOS RIOS PROVINCE

COTACANI PROVINCE

Map data ©2018 Google Terms of Use

**Figura 3.3 Registro de cliente en el sistema**

### 3.1.2. Perfiles de usuario

Los usuarios son los que van a acceder al sistema usando la autenticación, y los grupos o perfiles definirán a que función del sistema el usuario tendrá acceso. En este caso se creará un grupo que será para el personal técnico de la empresa, y podrá crear, modificar y administrar clientes y dispositivos de red, así como también podrá ver el estado actual e histórico de la red.

El uso del sistema estará dado por dos perfiles de usuarios, el usuario privilegiado que tendrá total control en la administración de los registros de cada una de las entidades del sistema, y es el único que creará usuarios y perfiles de acceso.

El segundo, es el técnico de red que será ingresado por el usuario privilegiado, podrá administrar la información del cliente, nombre, dirección, podrá registrar los datos de antenas, ubicación, marca, tipo de antena, y otros equipos de red con datos como dirección IP, interfaces y credenciales de acceso.

En la figura 3.4 se puede ver qué datos el usuario técnico puede subir, la información de dispositivos, los campos que no tienen asteriscos pueden almacenarse vacíos esto servirá para reconocer que tipo de acceso el equipo

tendrá. Se tiene la opción para seleccionar si el equipo está ligado a una antena o no, está hecho de esa manera para generalizar los dispositivos y no crear más información en la base de datos.

The screenshot shows a web form titled 'GENERAL HISTORIAL EQUIPOS'. The form contains the following fields and values:

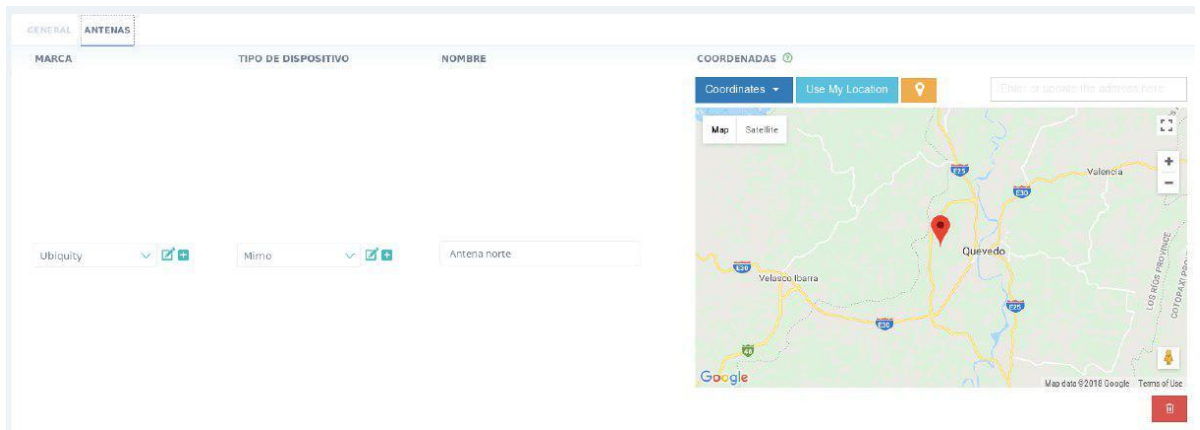
- Antena: Antena 1 (dropdown menu)
- Marca\*: Ubiquity (dropdown menu)
- Mac address\*: 00:0a:95:9d:68:16 (text input)
- Dirección IP o Hostname\*: 127.0.0.1 (text input)
- Ssh usuario: sflorenc (text input)
- Ssh contraseña: \*\*\*\*\* (password field)
- Ssh puerto: 22000 (text input)
- Ssh clave publica: (empty text input)
- Versión SNMP: 1 (text input)
- Snmp community: public (text input)
- Snmp usuario: (empty text input)
- Snmp contraseña: (empty password field)

At the bottom of the form, there are three buttons: 'Save', 'Save and add another', and 'Save and continue editing'.

**Figura 3.4 Registro de equipos**

### 3.1.3. Ubicación geográfica de antenas

Cada cliente puede tener instalada una o varias antenas, por lo que se creó una referencia al cliente en cada antena, además de guardar la ubicación de cada una de ellas. En la figura 3.5 podemos ver la pantalla de ingreso de antenas.



**Figura 3.5 Registros de antenas del cliente**

En la siguiente figura 3.6 se muestra un mapa listando todas las antenas que el usuario ingresó que han sido ubicadas según las coordenadas indicadas por el usuario en el sistema, en el mapa se va filtrará antenas por clientes, y se podrá ver el estado de cada una de las antenas.



**Figura 3.6 Mapa de antenas registradas**

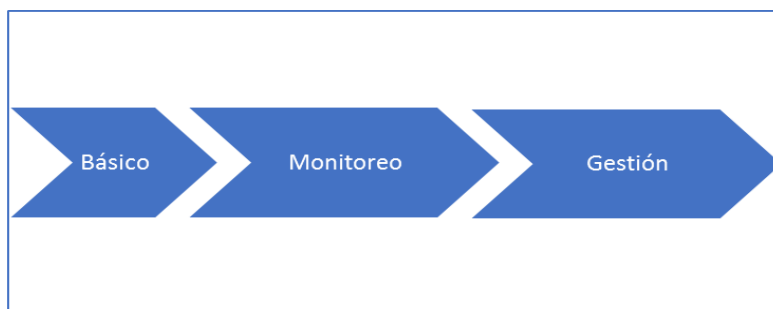
### 3.1.4. Monitoreo en tiempo real

FECHA	RT MIN	RT MAX	RT STDDEV	RT AVG	PACKET LOSS	TIME TOTAL	STATUS	DELETE?
Aug. 15, 2018, 1:50 a.m.	0.108	0.108	0.000	0.108	0%	0ms	is up	<input type="checkbox"/>
Aug. 15, 2018, 1:50 a.m.	0.051	0.051	0.000	0.051	0%	0ms	is up	<input type="checkbox"/>
Aug. 15, 2018, 1:50 a.m.	0.039	0.039	0.000	0.039	0%	0ms	is up	<input type="checkbox"/>
Aug. 15, 2018, 1:51 a.m.	0.054	0.054	0.000	0.054	0%	0ms	is up	<input type="checkbox"/>
Aug. 15, 2018, 1:51 a.m.	0.029	0.029	0.000	0.029	0%	0ms	is up	<input type="checkbox"/>

[+ Add another Historial Interfaz](#)

**Figura 3.7 Listado de escaneos realizados**

A partir del listado de todos los equipos ingresados en el sistema, se podrá realizar monitoreos en tiempo real de cada uno, como se puede ver en la figura 3.7, pero dependerá del nivel de acceso figura 3.8 que el usuario tenga en cada dispositivo, estos niveles serán explicados a continuación.



**Figura 3.8 Niveles de acceso hacia dispositivos de red**

#### 3.1.4.1. Acceso básico

El primero es el nivel básico, que recibirá información obtenida por un ping a partir de una dirección IP, no requiere ningún tipo de credencial para acceder al contenido del equipo, simplemente indicará en el sistema si la dirección IP es alcanzable; mediante el uso del protocolo ICMP que usa el comando Ping, se mostrará una lista con la información de la cantidad de los paquetes enviados, recibidos y perdidos, así como también los tiempos de ida y vuelta aproximados,



estos datos serán mostrados en el aplicativo web, con la fecha y hora en la que se ejecutó la orden.

Mediante la consulta de todas las interfaces que están registradas en el sistema, se ejecutará un script [Apéndice C] que va a obtener la información obtenida del comando Ping, luego de traducir los datos a un formato de diccionario se va a guardar en la base de datos para su posterior visualización. Este script será una acción adicional que la aplicación va a ejecutar, y gracias al uso de Django[1] los resultados de los comandos ejecutados serán actualizados en la base de datos.

#### **3.1.4.2. Acceso de monitoreo**

El segundo nivel el de monitoreo, que consiste en el uso del protocolo SNMP mediante la herramienta PySNMP[3], para ello se requerirá que el dispositivo a escanear tenga instalado el Agente SNMP y en el sistema agregar el código de comunidad para poder tener acceso.

Igual como en la sección anterior se va a monitorear a cada interfaz que exista en el sistema, siempre tomando en cuenta que tenga activada la opción de acceso de monitoreo, en este proyecto solo se van a usar la versión 1 de SNMP, puesto que cada versión es distinta, este script no es compatible con otras versiones SNMP.

Como resultado de las interfaces que están en cada equipo, tenemos: la descripción de la interface, el tipo, el tamaño en octetos del paquete más grande que es recibido o enviado por el interfaz denominado MTU, el ancho de banda actual en bits por segundo, el tipo de interface, MAC address, el estado deseado de la interfaz usando ifAdminStatus, el estado actual operativo de la interfaz. Importante aclarar que los dispositivos que se quieran analizar deben tener activado el protocolo SNMP, y el usuario del sistema debe haber ingresado la comunidad SNMP que sirve como autenticación hacia los equipos.

### **3.1.4.3. Acceso de gestión**

Este último nivel permite la gestión de los equipos mediante el acceso por SSH, para esto en el sistema el equipo debe tener registrado el nivel de gestión, se requieren tener las credenciales del equipo, se podrá consultar y modificar la configuración de los equipos de red, es importante también saber qué marca es de cada equipo porque los comandos varían dependiendo del fabricante y sistema operativo.

Se ha utilizado la librería Paramiko, que es una implementación de Python del protocolo SSH, permite crear scripts que simulan una conexión interactiva.

Ninguno de los niveles de acceso depende entre sí, sin embargo, se puede tener una combinación de estos, provisionando de mayor cantidad de información acerca de la red.

### **3.1.5. Notificación de alertas**

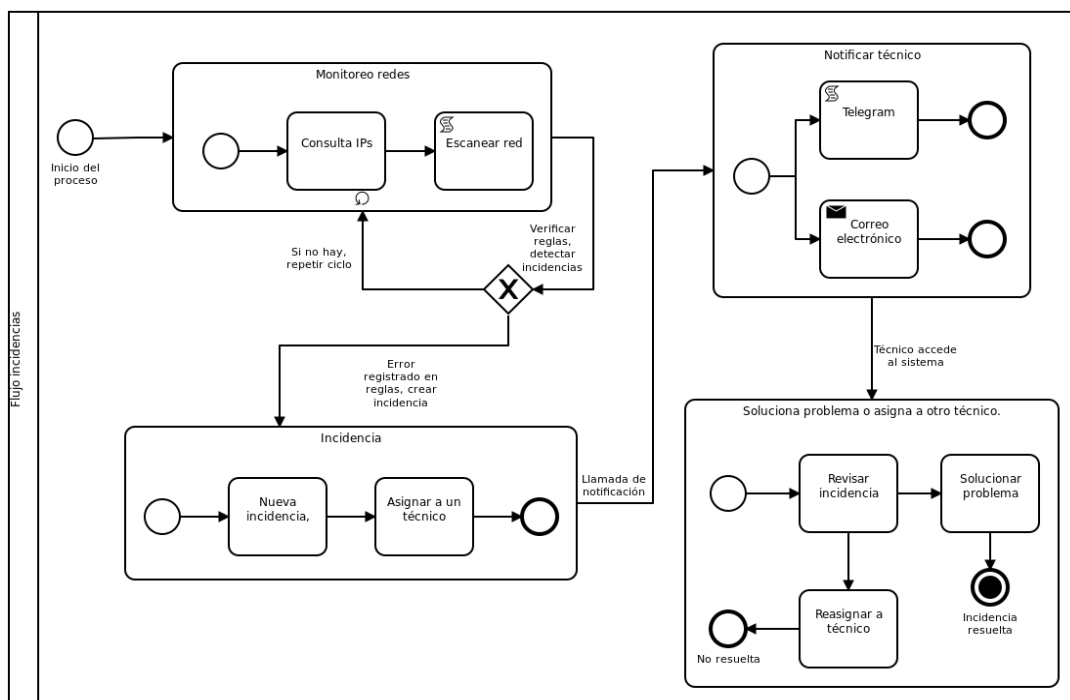
Para que existan notificaciones debe haber reglas establecidas por el usuario, estas se componen de un mensaje de error y valores máximos y mínimos que serán evaluadas cada vez que el sistema reciba datos efectuados por los escaneos a los distintos dispositivos, cuando exista un valor fuera del rango evaluado por las reglas, inmediatamente el bot de Telegram enviará un mensaje al usuario.

### **3.1.6. Gestión de errores, reglas, incidencias**

En el registro de errores, primero se debe contar con una serie de reglas que deben ser ingresadas por el usuario en el sistema, para que en el momento de recibir los resultados del escaneo a dada interfaz y estos valores no se encuentren dentro de los predeterminados, lo tomará como error y almacenará los valores y mensaje presentado, seguido de un envío de alerta al usuario.

Para que el sistema detecte errores es necesario haber registrado reglas en el sistema previamente que serán consultadas siempre que se realicen escaneos a los dispositivos, al ser detectado un error se envía un mensaje al usuario técnico, y automáticamente se genera una incidencia en el sistema, dónde el usuario notificado previamente ingresará al sistema y se auto-asignará la incidencia o lo derivará a otro usuario, a partir de eso el usuario encargado tendrá que resolver el problema y cuando lo haga, reportará la solución en la incidencia y procederá a seleccionar el estado en caso de resolverlo.

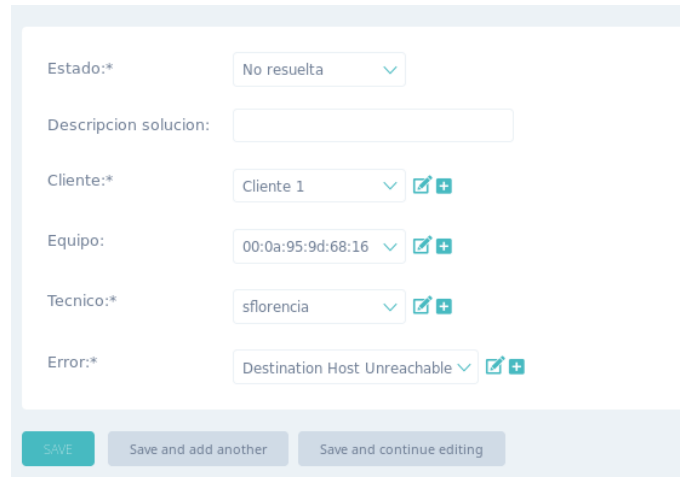
En la figura 3.9 se observa el flujo del desarrollo de un reporte de incidencias a partir de la detección de un error en el análisis de la red.



**Figura 3.9 Proceso de notificación de errores y asignación de incidencia**

Para reportar la incidencia se registrarán los datos mostrados en la figura 3.10, se almacenará el error obtenido, los datos del equipo y el cliente, el estado actual de esa incidencia, además se asignará a un encargado que ha sido notificado para que este proceda a solucionar esa incidencia, que una vez resuelta deberá

registrarla como resuelta y colocar la solución que aplicó en ella, en caso de que el técnico no pueda dar solución, tendrá que asignar la incidencia a otro técnico.



The image shows a web form for registering an incident. It contains several fields with dropdown menus and a text input field. The fields are: 'Estado:\*' with a dropdown set to 'No resuelta'; 'Descripcion solucion:' with a text input field; 'Cliente:\*' with a dropdown set to 'Cliente 1'; 'Equipo:' with a dropdown set to '00:0a:95:9d:68:16'; 'Tecnico:\*' with a dropdown set to 'sflorenca'; and 'Error:\*' with a dropdown set to 'Destination Host Unreachable'. Each dropdown menu has a small icon of a document with a plus sign. At the bottom of the form, there are three buttons: 'SAVE' (highlighted in green), 'Save and add another', and 'Save and continue editing'.

**Figura 3.10 Ventana de registro de incidencias**

### 3.1.7. Reportes

Los reportes generados estarán basados en la información ingresada por el usuario y los resultados de los scripts generados. A continuación, se listan los reportes que la aplicación tendrá:

- Clientes registrados y las antenas que posee.
- Escaneos ejecutados a cada uno de los dispositivos.
- Reglas configuradas para detectar errores.
- Incidencias creadas en base a la detección de errores.

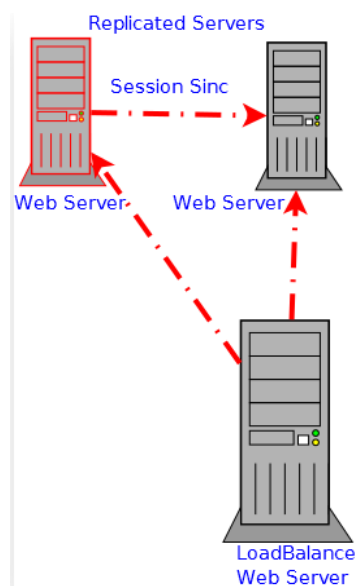
El reporte de incidencias consiste en un listado de todas las incidencias ocurridas en alguna de los equipos escaneados, los campos que han sido tomados en cuenta para este reporte son los mostrados en la figura 3.11. Pueden ser filtrados por técnico, estado, fecha.

	A	B	C	D	E	F
1	fecha	Error	Solución	Técnico encargado	Estado	
2	"15/08/2018 08:00:00"	Host destino inalcanzable	Reinicio de equipo	Sanny Florencia	Atendido	
3	"21/08/2018 15:30:45"	Host destino inalcanzable			Asignando técnico	
4	"21/08/2018 18:12:10"	Request timed out (Request Timeout)	Cable desconectado	Sanny Florencia	Atendido	
5						
6						
7						

**Figura 3.11 Reporte de incidencias**

### 3.1.8. Sistema de alta disponibilidad y balanceo de carga

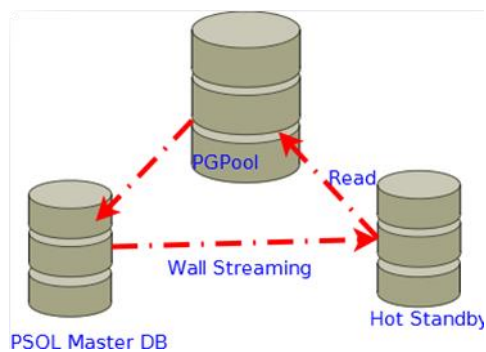
Brindar soporte 24/7 es muy importante asegurar al cliente monitorear sus redes y dar solución inmediata a las incidencias que ocurran. Se presenta la siguiente arquitectura en la figura 3.12, que consta de 3 servidores Ubuntu 18.04, el primer servidor es que el balanceara la carga, va a detectar fallas en alguno de los componentes de los servidores de replicación y adaptará el sistema al que no tenga fallas.



**Figura 3.12 Arquitectura de alta disponibilidad y balanceo de carga.**

### 3.1.9. Base de datos tolerante a fallas

Es importante que exista un sistema de bases de datos que sea resistente a fallas, es por eso se diseñó una arquitectura mostrada en la figura 3.13, para el servicio de FailOver, LoadBalancing, y Replicación en la base de datos, con el uso de la herramienta PGPool[7] que es un software intermediario entre el sistema operativo Ubuntu y la base de datos PostgreSQL.



**Figura 3.13 Base de datos Failover**

# CAPÍTULO 4

## 4. IMPLEMENTACIÓN Y PRESUPUESTO

### 4.1 Plan de implementación

Este cronograma está considerado para la implementación de la aplicación web, y consta de 80 días a partir del acuerdo entre las dos partes.

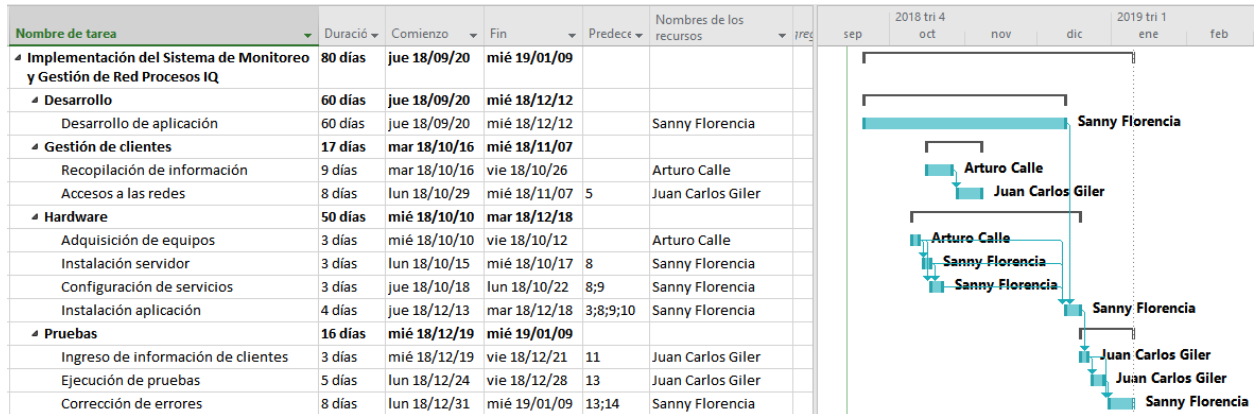


Figura 4.1 Plan de implementación

### 4.2 Presupuesto

Desarrollo para la implementación de la aplicación de monitoreo de redes es el siguiente.

Tabla 4.1 Presupuesto implementación

Cant	Descripción	Tiempo (meses)	Valor (mes)	Total
3	Servidores Digital Ocean RAM 4 GB 2 vCPUs SSD 80 GB	24	\$20.00	\$480.00
1	Mano de obra	1	\$5,000.00	\$5,000.00
1	Técnico de red	3	\$1,000.00	\$3,000.00
2	Programadores	2	\$1,000.00	\$4,000.00
1	Mantenimiento	24	\$560.00	\$1,344.00
1	Google Maps API	24	\$30.00	\$720.00
	<b>Total</b>			<b>\$26,640.00</b>

# CONCLUSIONES Y RECOMENDACIONES

## Conclusiones

- No fue posible desarrollar un solo tipo de script para monitorear los equipos de diferentes marcas, por ello se necesitará de submódulos por cada una de las marcas con las que el cliente trabaje.
- Para acceder un equipo es necesario contar con las credenciales de administración de cada uno de ellos, estos datos son sensibles y es necesario proteger la información.
- Con el servicio SNMP, los equipos a monitorear deben activar el protocolo en su sistema operativo, y dependiendo de la versión se necesita obtener los códigos de comunidad y credenciales.
- En caso de no contar con credenciales para acceder a los equipos, se puede obtener información más básica sobre estos, es necesario tener una dirección IP pública para realizar pruebas de ping.

## Recomendaciones

- Para poder medir los tiempos de respuesta de cada incidente y evaluar el tiempo que toma responder cada solicitud es estrictamente necesario establecer reglas de funcionamiento en el cual se van a especificar los mínimos y máximos que pudiera resultar de la ejecución de un comando.
- Las pruebas que se ejecutan sobre los dispositivos son totalmente remotas, a través del servidor que se encuentra alojado en la nube, pero para esto se deben registrar en el sistema todos los dispositivos que quieren ser revisados.



- Dado que existen una gran cantidad de comandos para el monitoreo de redes se recomienda realizar una nueva interfaz para el ingreso de nuevos comandos y que se ejecuten automáticamente en el sistema.
- Debido a que el acceso de administración de un equipo puede ser limitado, se deben crear tres niveles de monitoreo que van desde el cálculo de latencia por ICMP a partir de una dirección IP pública, que se usará de manera exclusiva en dispositivos sin SNMP activado y a los que no se tiene credenciales de administración.

# BIBLIOGRAFÍA

[1] Django Software Foundation (2018) Framework Django [Online]. Disponible en:  
<https://docs.djangoproject.com/en/2.1/intro/tutorial01/>

[2] Google Maps Platform (2018, Julio, 17) Google Maps [Online]. Disponible en:  
<https://developers.google.com/maps/documentation/javascript/examples/map-simple>

[3] SnmpLabs (2018) Example scripts [Online]. Disponible en:  
<http://snmplabs.com/pysnmp/examples/contents.html>

[4] Django Software Foundation (2018) Custom Commands Django [Online].  
Disponible en:  
<https://docs.djangoproject.com/en/2.1/howto/custom-management-commands/>

[5] PyMOTW-3 (2018, Marzo, 18) Python Subprocess [Online]. Disponible en:  
<https://pymotw.com/3/subprocess/>

[6] Apache Foundation (2018) Apache Module mod\_proxy\_balancer [Online].  
Disponible en:  
[https://httpd.apache.org/docs/2.4/mod/mod\\_proxy\\_balancer.html](https://httpd.apache.org/docs/2.4/mod/mod_proxy_balancer.html)

[7] PGPool (2018, Agosto, 20) Failover and Failback [Online]. Disponible en:  
<http://www.pgpool.net/docs/latest/en/html/runtime-config-failover.html>

# **ANEXOS**



# ANEXO A

## Metodología Design Thinking Entrevistas

ESCUELA SUPERIOR POLITECNICA DEL LITORAL

LICRED

Es variable, existen productores que ~~se~~ le interesa el acceso solo a las oficinas, otros toda la hacienda, otros ~~solo~~ todos los trabajadores 20 o 30 personas concurrentes.

6. ¿Con cuáles ISPs trabajan?

Actualmente con Telconet, solo ellos llegan a los sectores de los clientes.

7. ¿Cuál es el tiempo de respuesta aproximado del soporte técnico del ISP?

Si varía, pero es decente. Usualmente no ocurren problemas pero cuando se rompe la fibra o se han robado, pueden pasar días.

8. ¿Cuántas tienen un enlace dedicado al ISP?

Un enlace dedicado x cliente

• Ofrecen Soporte 24/7?

No solo horarios de oficina.

9. ¿En qué sector se encuentran?

En los ríos <sup>cerca de</sup> sector Quevedo.

10. ¿Los equipos de red reciben mantenimiento continuo?

Si se les da mantenimiento pero no de forma planificada

11. ¿Existe algún encargado de IT en las haciendas?

En haciendas pequeñas no. En haciendas grandes si hay personal IT pero no son dependiente de procesos IA. El personal de IT de la empresa es de Guayaquil.

12. ¿Cuál es el tiempo de respuesta de la empresa a los problemas de red?

Es bastante alto porque tienen que viajar de Gye - Quevedo para conectarse a los equipos puede durar 6 horas. Si es cambio de direccion de antenas puede ser 2 dias.

13. ¿Cuentan con un registro histórico a problemas comunes?

No existe un registro <sup>formal</sup>, solo es de forma mantenimiento. Solo cuentan con la experiencia del Ing.

14. ¿Cuentan con un plan preventivo ante problemas comunes?

No existe en absoluto un plan preventivo. Se hace de forma reactiva. Se espera a que el cliente informe para realizar la inspeccion.

15. ¿Una vez solucionado el problema, se realizan pruebas periódicas para ver el estado de la red?

Si se realiza un conjunto de pruebas para asegurar que todo esté ok! Pero ~~no~~ lo hacen solo después de haber solucionado el problema.

\* Dato importante!

o El servicio de internet como tal es directamente  
 Productor — Telconet (contratan enlace dedicado)

Procesos IQ a partir de ese enlace arma la infraestructura para conectar las otras haciendas.

o Antes los productores solo tenían ese enlace y solo la hacienda instalada tenía internet

## ANEXO B

### ***Aplicación web***

La aplicación web necesita tener una base de datos en donde va a almacenar toda la información de sus registros, es por ello que lo primero a realizar será la creación de la base de datos usando PostgreSQL para esto tenemos la sentencia:

```
CREATE DATABASE NombreDB;
```

A continuación, lo siguiente será crear la aplicación web, por esto se ha decidido usar el Framework denominado Django que utiliza Python como su lenguaje de programación. Comenzamos con la creación de un nuevo proyecto:

```
$ django-admin startproject mysite
```

Luego se crean aplicaciones dentro del proyecto, aquí es donde se va a crear los modelos y las vistas de la página web:

```
$ python manage.py startapp web
```

A partir de los criterios y requerimientos que tomaron para la definición de la solución se crean los modelos que son todas las entidades que tendremos en la base de datos.

Luego de tener los modelos en la aplicación se deben crear en la base de datos, pero primero se debe configurar la base de datos en la aplicación, dirigiéndose al archivo ***mysite/settings.py*** una vez configurado se podrán crear las tablas a partir de los modelos en la aplicación con los siguientes comandos:

```
$ python manage.py makemigrations web
```

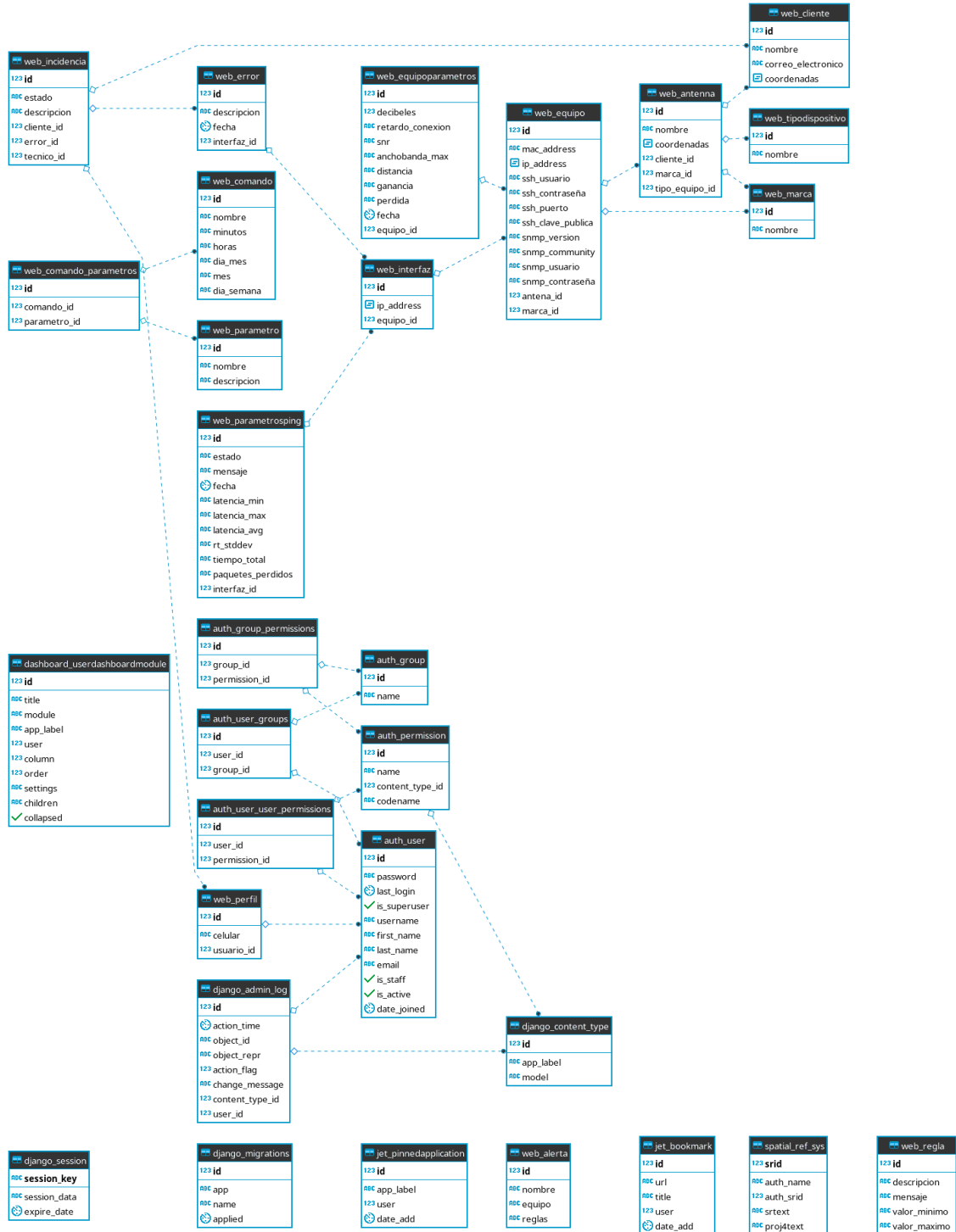
```
$ python manage.py migrate
```

Como la aplicación y la base de datos ya se comunican entre sí se va a proceder a crear vistas para visualizar las tablas, la ventaja de usar Django es que ya viene con un panel de administrador en el cual se van a indicar cuales son los modelos que se quieren mostrar y como se van a mostrar.



Se va a generar una vista adicional en la que se mostrará un mapa ubicando antenas en él, para esto se usa el API de Google Maps que va a permitir a partir de un mapa manipularlo para agregar en los marcadores según las coordenadas almacenadas en los modelos.

# ANEXO C



## **ANEXO D**

Los scripts que permiten monitorear redes usando el comando Ping, los módulos PySNMP y Paramiko, para luego almacenar los datos dentro de la aplicación serán desarrollados utilizando una herramienta de Django denominada Custom Management Commands, donde la aplicación registrará acciones mediante el comando *django manage.py*. Además, para la ejecución de comandos de Linux se usará la herramienta Subprocess que permitirá invocar un proceso a través de nuestro script en Python.

```

from django.core.management.base import BaseCommand, CommandError
from web.models import Interface, InterfacePingParams
from django.utils import timezone
import subprocess
from . import app
class Command(BaseCommand):
    help = 'Check availability of network'
    def handle(self, *args, **options):
        interfaces = Interface.objects.all()
        for i in Interface.objects.all():
            count='1'
            wait_sec='1'
            server = i.ip_address
            cmd = ["ping", "-c", count, "-W", "wait_sec", server]
            try:
                completed = subprocess.run(cmd, check=True, stdout=subprocess.PIPE,)
                output = completed.stdout.decode('utf-8')
                total, loss, timing = "", "", ""
                for l in output.split("\n"):
                    if 'packets transmitted' in l:
                        total = l.split(',')[3].split()[1]
                        loss = l.split(',')[2].split()[0]
                    elif 'Host':
                        msj = l.split(':')[1]
                    elif 'rtt' in l:
                        timing = l.split()[3].split('/')[0]
                ping = InterfacePingParams(
                    interface = i,
                    date = timezone.now(),
                    rt_min = timing[0],
                    rt_avg = timing[1],
                    rt_max = timing[2],
                    rt_stddev = timing[3],
                    time_total = total,
                    msj = msj,
                    packet_loss = loss
                )
                ping.save()
                check_ping = Regla.objects.filter(mensaje=ping.msj, minimo<ping.rt_min, maximo>ping.rt_max)
                if check_ping:
                    app.send_alert(ping)
            except subprocess.CalledProcessError as err:
                print('Error',err)

```

## Script 1 Escaneo vía ping

```

from django.core.management.base import BaseCommand, CommandError
from web.models import Interface, InterfacePingParams
from django.utils import timezone
import subprocess
from . import app
from pysnmp.hlapi import *
class Command(BaseCommand):
    help = 'Check availability of network'
    def handle(self, *args, **options):
        interfaces = Interface.objects.all()
        for i in Interface.objects.all():
            for (errorIndication,
                errorStatus,
                errorIndex,
                varBinds) in nextCmd(SnmpEngine(),
                                    CommunityData('public', mpModel=0),
                                    #UdpTransportTarget((i.ip_address, 161)),
                                    UdpTransportTarget(('demo.snmplabs.com', 161)),
                                    ContextData(),
                                    ObjectType(ObjectIdentity('IF-MIB', 'ifDescr')),
                                    ObjectType(ObjectIdentity('IF-MIB', 'ifType')),
                                    ObjectType(ObjectIdentity('IF-MIB', 'ifMtu')),
                                    ObjectType(ObjectIdentity('IF-MIB', 'ifSpeed')),
                                    ObjectType(ObjectIdentity('IF-MIB', 'ifPhysAddress')),
                                    ObjectType(ObjectIdentity('IF-MIB', 'ifType')),
                                    lexicographicMode=False):
                if errorIndication:
                    print(errorIndication)
                    break
                elif errorStatus:
                    print('%s at %s' % (errorStatus.prettyPrint(),
                                        errorIndex and varBinds[int(errorIndex)-1][0] or '?'))
                    break
                else:
                    for varBind in varBinds:
                        print(' = '.join([x.prettyPrint() for x in varBind]))

```

### Script 3 Escaneo vía SNMP

