

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE UNA HERRAMIENTA DE SOFTWARE PARA EL ANÁLISIS DE VULNERABILIDADES EN DISPOSITIVOS IOT EN ECUADOR.

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN ANÁLISIS DE SISTEMAS INFORMÁTICOS

EDISON STALIN JUMBO JUMBO

edison.jumbo@epn.edu.ec

JEFFERSON RICARDO LLUMIQUINGA LLUMIQUINGA

jefferson.llumiquinga@epn.edu.ec

DIRECTOR: DR. RICHARD PAUL RIVERA GUEVARA

richard.rivera01@epn.edu.ec

CODIRECTOR: DRA. MYRIAM GUADALUPE PEÑAFIEL AGUILAR

myriam.peñafiel@epn.edu.ec

Quito, septiembre 2021

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por los Sres. Jumbo Jumbo Edison Stalin y Llumiquinga Llumiquinga Jefferson Ricardo como requerimiento parcial a la obtención del título de TECNÓLOGO EN ANÁLISIS DE SISTEMAS INFORMÁTICOS, bajo nuestra supervisión:

Dr. Richard Rivera

DIRECTOR DEL PROYECTO

Dra. Myriam Guadalupe

CODIRECTOR(A) DEL PROYECTO

DECLARACIÓN

Nosotros Jumbo Jumbo Edison Stalin con CI: 1725303711 y Llumiquinga Llumiquinga Jefferson Ricardo con CI: 1721130993 bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 144 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación – COESC-, somos titulares de la obra en mención y otorgamos una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional.

Entregamos toda la información técnica pertinente, en caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.

DEDICATORIA

Dedico este proyecto a mis padres quienes me han brindado apoyo incondicional a lo largo de toda mi carrera profesional y por quienes sigo cumpliendo mis objetivos. A mis hermanos y todas las personas, que me han ayudado y han hecho que le presente trabajo se culmine exitosamente en especial a quienes me han brindado sus conocimientos y fuerza en los momentos difíciles.

EDISON STALIN JUMBO JUMBO

AGRADECIMIENTO

Primero agradezco a Dios, por haberme permitido cumplir con una de las metas que me he propuesto en mi vida.

Agradezco a mis padres, Merci Jumbo y Jofre Erique, quienes me apoyaron incondicionalmente con sus consejos y motivación constante.

A mis hermanos Gabriela, María y Jefferson, quienes siempre me motivaron a continuar con mis objetivos y de quienes soy un ejemplo a seguir.

A mis amigos, quienes fueron un apoyo incondicional en los momentos de difícil decisión, con quienes también he pasado momentos inolvidables.

A todos y cada uno de los ingenieros que me impartieron sus conocimientos durante mi formación académica, en especial al Dr. Richard Rivera, por sus consejos, comprensión y paciencia en el desarrollo de este proyecto.

EDISON STALIN JUMBO JUMBO

DEDICATORIA

El presente proyecto va dedicado a mis padres y hermano ya que, gracias a sus esfuerzos, amor, sacrificio y apoyo incondicional e llegado a culminar mi carrera universitaria, a las personas que con sus conocimientos me han ayudado para que el presente trabajo sea culminado de manera exitosa.

JEFFERSON RICARDO LLUMIQUINGA LLUMIQUINGA

AGRADECIMIENTO

Agradezco principalmente a mis padres que nunca dejaron de confiar en mí y siempre me estuvieron apoyando a lo largo de mi carrera universitaria ya que gracias a ellos he luchado para cumplir con el objetivo que tuve desde el primer día que llegué a la Escuela Politécnica Nacional.

A Cindy Viracucha que ha estado en las buenas y malas conmigo y que con su cariño y apoyo incondicional llegue a encontrar el camino correcto y así poder culminar con éxito mi carrera universitaria.

A todo mi círculo de amistades que siempre han estado ahí conmigo en las buenas y en las malas brindándome un apoyo necesario para no desfallecer.

A mis amigos y amigas que encontré en la universidad siempre siendo esas buenas amistades.

A todos los ingenieros/as que compartieron su conocimiento conmigo desde el primer semestre hasta el día de hoy, de manera especial agradezco a mi tutor el Dr. Richard Rivera por su dedicación y ayuda brindada para poder culminar con éxito este trabajo de titulación.

JEFFERSON RICARDO LLUMIQUINGA LLUMIQUINGA

ÍNDICE DE CONTENIDOS

1.	Introducción	1
1.1	Objetivo General.....	2
1.2	Objetivos Específicos	2
1.3	Alcance.....	2
1.4	Estructura del documento	3
2	Metodología.....	3
2.1	Metodología de Desarrollo Scrum.....	3
2.1.1	Roles centrales.....	4
2.1.2	Artefactos	4
2.2	Diseño de interfaces del sistema	7
2.2.1	Marvel.....	7
2.3	Diseño de la arquitectura del sistema	7
2.3.1	Patrón arquitectónico.....	8
2.3.2	Sistema Web	8
2.4	Herramientas de desarrollo.....	8
2.4.1	Librerías	9
3.	ResultadoS y discusion.....	10
3.1	Sprint 0. Configuración del ambiente de desarrollo.....	11
3.1.1	Elaboración de la Base de datos en MongoDB.....	11
3.1.2	Estructura del proyecto.....	12
3.1.3	Roles del usuario.....	13
3.1.4	Requerimientos para el <i>Script</i> de búsqueda y almacenamiento de direcciones IPv4	14
3.1.5	Requerimientos para el Sistema Web.....	15
3.2	Sprint 1. Buscar direcciones IPv4	15

3.2.1	Buscar direcciones IPv4	16
3.2.2	Actualizar direcciones IPv4.....	16
3.2.3	Capturar pantalla de puertos abiertos.....	17
3.3	Sprint 2. Recolección de datos	17
3.3.1	Recolectar direcciones IPv4	17
3.4	Sprint 3. Inicio de sesión del usuario administrador	19
3.4.1	Iniciar y cerrar sesión de Administrador.....	19
3.5	Sprint 4. Inicio de sesión del usuario cliente	20
3.5.1	Inicio de sesión con rol usuario.....	21
3.5.2	Registro de usuario cliente	22
3.5.3	Editar Perfil.....	23
3.5.4	Visualización de página de inicio	24
3.5.5	Visualizar direcciones IPv4.....	25
3.5.6	Buscar direcciones IPv4	25
3.5.7	Análisis de direcciones y puertos encontrados	27
3.6	Sprint 5. Pruebas de software.....	27
3.6.1	Pruebas funcionales	27
3.7	Discusión.....	28
4.	Conclusiones y Recomendaciones	33
4.1	Conclusiones.....	33
4.2	Recomendaciones.....	33
5.	Referencias	34
6.	Anexos	39
6.1	Manual Técnico	39
6.2	Manual de Usuario.....	39
6.3	Manual de Instalación.....	39

ÍNDICE DE FIGURAS

Fig. 1: Visualizar direcciones IPv4.....	7
Fig. 2: Patrón Arquitectónico	8
Fig. 3: Colección en MongoDB.....	11
Fig. 4: Direcciones IPv4 almacenadas	12
Fig. 5: Script de búsqueda y recolección de direcciones IPv4	12
Fig. 6: Estructura del Sistema Web	13
Fig. 7: Script Python.....	13
Fig. 8: Activación de cuentas de usuarios	14
Fig. 9: Sistema Web.....	14
Fig. 10: Menú de opciones	16
Fig. 11: Cantidad de direcciones IPv4 a buscar.....	16
Fig. 12: Actualización de direcciones IPv4	17
Fig. 13: Captura de pantalla	17
Fig. 14: Recolección de direcciones IPv4	18
Fig. 15: Recolección de datos en MongoDB.....	18
Fig. 16: Índices del documento Devices	19
Fig. 17: Inicio de sesión rol administrador	20
Fig. 18: Panel para activar cuentas	20
Fig. 19: Inicio de sesión rol usuario	21
Fig. 20: Información en el Dashboard.....	22
Fig. 21: Registro de usuario final.....	22
Fig. 22: Editar Perfil.....	23
Fig. 23: Editar contraseña	23
Fig. 24: Mapa cantidad de puertos por provincia	24
Fig. 25: Aparatos IoT e información de desarrolladores	24
Fig. 26: Visualización de direcciones IPv4.....	25
Fig. 27: Búsqueda por la opción de dirección.....	25
Fig. 28: Búsqueda por la opción de puerto	26
Fig. 29: Búsqueda por la opción de ciudad.....	26
Fig. 30: Análisis de puertos por ciudad.....	27
Fig. 31: Cantidad de direcciones IPv4 en MongoDB.....	29
Fig. 32: Cantidad de datos con puertos abiertos	29

Fig. 33: Total de capturas en el directorio.....	29
Fig. 34: Numero de direcciones IPv4 con puerto 80	30
Fig. 35: Captura de pantalla en puerto 80	30
Fig. 36: Puerto 80 en Pichincha	31
Fig. 37: Puerto 443 en Pichincha.....	31
Fig. 38: Puerto 22 en Guayas.....	32
Fig. 39: Puerto 80 en Guayas.....	32

ÍNDICE DE TABLAS

TABLA I: Asignación de roles en el equipo de trabajo.....	4
TABLA II: Historia de Usuario 1 – Iniciar sesión de usuario	5
TABLA III: Elaboración de Product Backlog	5
TABLA IV: Herramientas para el desarrollo del Sistema Web y Script	9
TABLA V: Librerías utilizadas para el desarrollo del Sistema Web y Script.....	10
TABLA VI: Pruebas de aceptación - Ingreso al Sistema Web	28
TABLA VII: Navegadores utilizados para la ejecución de la prueba	28

RESUMEN

El presente informe técnico presenta el desarrollo de una herramienta de software para el análisis de vulnerabilidades en dispositivos IoT en Ecuador. El proyecto se desarrolla con el fin de realizar un análisis de vulnerabilidades acerca de direcciones IPv4 y puertos en dispositivos conectados a Internet, a través de la herramienta es realizada la búsqueda de direcciones IPv4 que están dentro del rango del territorio ecuatoriano, se observan los puertos posibles a estar activos o abiertos en cada dirección IPv4.

La herramienta se divide en 2 partes la primera es la creación del *Script* de búsqueda de direcciones IPv4 la cual es desarrollado en su totalidad en *Python*, un lenguaje de programación multiplataforma de código limpio y versátil que además cuentan con un sin número de librerías que se van actualizando continuamente, por otra parte el desarrollo del Sistema Web permite visualizar, buscar y detallar las direcciones IPv4 ha sido desarrollado mediante *Flask* el cual es un *Microframework* desarrollado en *Python* y que permite trabajar con el patrón MODELO, VISTA , CONTROLADOR(MVC).

Los resultados obtenidos durante el proceso de búsqueda de direcciones IPv4 han sido satisfactorios, encontrando información como: cámaras web, servidores Apache, páginas web e incluso petición de credenciales para acceder a la configuración de un router, con estos resultados se refleja que el nivel de seguridad informática en el Ecuador se encuentra en un nivel bajo. Se tiene en cuenta que en Pichincha y Guayas al ser las dos provincias más grandes del Ecuador tienen el mayor número de direcciones IPv4 con puertos activos.

PALABRAS CLAVE: Script, Flask, Python, Sprint, IPv4, IoT-Ecuador

ABSTRACT

This technical report presents the development of a software tool for the analysis of vulnerabilities in IoT devices in Ecuador. The project is developed in order to perform a vulnerability analysis about IPv4 addresses and ports in devices connected to the Internet, through the tool the search for IPv4 addresses that are within the range of the Ecuadorian territory is carried out, the ports are observed possible to be active or open on each IPv4 address.

The tool is divided into 2 parts, the first is the creation of the IPv4 address search script, which is developed entirely in Python, a multiplatform programming language with clean and versatile code that also has a number of libraries that can be used. They are continuously updating, on the other hand the development of the Web System allows to visualize, search and detail the IPv4 addresses has been developed using Flask which is a microframework developed in Python and that allows working with the MODEL, VIEW, CONTROLLER (MVC) pattern.

The results obtained during the IPv4 address search process have been satisfactory, finding information such as: webcams, Apache servers, web pages and even requesting credentials to access the configuration of a router, with these results it is reflected that the level of Computer security in Ecuador is at a low level. It is taken into account that in Pichincha and Guayas, being the two largest provinces in Ecuador, they have the highest number of IPv4 addresses with active ports.

KEYWORDS: *Script, Flask, Python, Sprint, IPv4, IoT-Ecuador*

1. INTRODUCCIÓN

El internet de las cosas (Internet of Things, en adelante IoT) hace referencia a dispositivos u objetos comunes que tienen la capacidad de conectarse a internet [1]. Durante los últimos años, se ha establecido la tendencia de tener cada vez más dispositivos interconectados a través de internet, con el objetivo de poder realizar de forma remota y/o automática acciones que permitan controlar tareas que se llevan a cabo de forma manual como el acceso a domicilios, control de luces, entre otros [2]. Esto ha provocado una gran demanda de fabricación de dispositivos inteligentes con acceso a Internet, más conocidos como “dispositivos IoT”. Un estudio realizado por IoT Analytics [3] indica que en el 2018 existían alrededor de 7 mil millones de dispositivos IoT conectados en el mundo, y se espera que para el 2025 la cantidad de dispositivos IoT conectados aumente a 22 mil millones.

Actualmente, las funciones principales de los dispositivos IoT son captar, controlar, procesar y almacenar los datos, los dispositivos IoT suelen ser cada vez más pequeños [4] [5], por lo que dependen de las puertas de enlace para poder comunicarse y ejecutar sus funciones [6]. Sin embargo, los dispositivos inteligentes desde un inicio no fueron diseñados para conectarse a Internet y esto provoca que sean más propicios a ciberataques; un ejemplo son los sistemas de control industrial (SCADA), que fueron diseñados para situarse en redes aisladas y actualmente se conectan a Internet [7]. Los mismos fabricantes de hardware se encargan del mantenimiento del software de los dispositivos IoT y gran parte de los fabricantes no disponen de la experiencia o recursos para poder responder ante posibles brechas de seguridad [8] o en algunos casos la seguridad no es una prioridad para ellos [9], [10].

Por otro lado, los usuarios son conscientes de los problemas de seguridad que afectan a los dispositivos IoT, y así lo demuestra una encuesta realizada por Enjoy Safer Technology (ESET), donde el 70% de los participantes considera que este tipo de dispositivos no son seguros, fundamentalmente en términos de privacidad, que es donde radica la principal preocupación. Sin embargo, el 62% considera que no dejara de comprar este tipo de tecnología [11].

En Ecuador los dispositivos IoT todavía no se han posicionado masivamente como en otros países de Europa [12], Asia o América del Norte. Algunas operadoras móviles y empresas privadas ya están empezando a ofrecer algunos servicios que involucran a dispositivos IoT [5], sin tomar en cuenta, aspectos de seguridad [13] y privacidad con

respecto a los servicios que ofrecen y la información que procesan. En la ciudad de Quito con el inicio de operaciones del Metro se espera que la cantidad de dispositivos IoT se incremente en gran medida [5], lo cual puede conllevar a un incremento de riesgos de la seguridad de la información, si estos dispositivos no consideran la seguridad como un aspecto fundamental de su despliegue.

Este proyecto tiene como objetivo desarrollar una herramienta que permita realizar un análisis de seguridad a dispositivos IoT conectados a Internet en el Ecuador, con la finalidad de obtener información para conocer las vulnerabilidades más comunes y problemas de seguridad a los que se enfrenta el despliegue de dispositivos IoT en Ecuador.

1.1 Objetivo General

Desarrollar un software para el análisis de vulnerabilidades de dispositivos IoT desplegados en Ecuador.

1.2 Objetivos Específicos

- Estructurar los requerimientos internos de sistema.
- Diseñar la arquitectura del sistema y base de datos.
- Desarrollar el sistema basado en los requerimientos.
- Probar el funcionamiento del sistema.

1.3 Alcance

Actualmente en Ecuador la cantidad de personas que cuentan con acceso a Internet es de más de 13.8 millones de usuarios, lo que significa que alrededor del 80% de la población ecuatoriana cuenta con internet [14]. Eso quiere decir que cada año la cantidad de usuarios conectados crece con respecto al anterior, permitiendo que los usuarios adopten nuevas tecnologías como es el Internet de las Cosas siendo uno de los propósitos la de automatizar el funcionamiento de cada uno de los dispositivos por medio de la interconectividad, mejorando la condición de las personas.

Como bien conocemos los teléfonos inteligentes son una muestra clara del uso de esta tecnología IoT en Ecuador ya que cuenta con varios sensores que permiten recolectar gran cantidad de información para posteriormente ser evaluada por un sistema inteligente computarizado, permitiendo tomar la mejor decisión de acuerdo con su propósito.

Ahora bien, una de las barreras que ha frenado esta tecnología en todo el mundo es la seguridad y privacidad de la información. Por ello, las industrias que fabrican estos dispositivos actualmente tienen como prioridad garantizar la seguridad del producto ante posibles vulnerabilidades. Debido que, al existir una pequeña falla de seguridad en el dispositivo, puede servir como una potencial entrada para ataques cibernéticos [15].

Por ello, este proyecto propone una herramienta que permita realizar un análisis de seguridad a dispositivos IoT conectado a Internet en el Ecuador, con la finalidad de obtener información para conocer las vulnerabilidades más comunes y problemas de seguridad a los que se enfrenta el despliegue de dispositivos IoT en Ecuador.

1.4 Estructura del documento

Este documento se encuentra estructurado de la siguiente manera: la sección I, corresponde a la introducción se da a conocer el contexto del problema, objetivo general, objetivos específicos y el alcance del proyecto. La sección II describe como se ha implementado la metodología, diseño de arquitectura y herramientas de desarrollo. La sección III muestra los resultados obtenidos en cada tarea realizada por Sprint. Finalmente, la sección IV se menciona las conclusiones y recomendaciones que se han obtenido a lo largo del desarrollo de la herramienta.

2 METODOLOGÍA

Las metodologías ágiles son indispensables dentro del desarrollo de software [16] ya que aportan eficiencia, calidad, flexibilidad y una pronta y efectiva respuesta con los involucrados en el proyecto.

La herramienta IoT se desarrolla mediante la metodología *Scrum* la cual parte con un levantamiento de requerimientos, siendo esta la tarea más importante para determinar la necesidad de la herramienta, determinar historias de usuario, *Product Backlog* y *Sprints* que nos garantizan la calidad del producto.

2.1 Metodología de Desarrollo Scrum

Scrum es una metodología de desarrollo ágil y flexible la cual permite abordar proyectos complejos desarrollados dentro de un entorno dinámico y cambiante, este tipo de metodología permite que los equipos de trabajo se organicen en base a las experiencias, todo esto se da bajo una serie de herramientas y recursos que ayudan al equipo para poder organizarse y se desempeñen con mayor agilidad [17].

2.1.1 Roles centrales

Esta metodología, conlleva el entender los roles y responsabilidades que son definidos en un proyecto y que al momento de entregar un producto o servicio se satisfaga los requisitos del cliente. Scrum está definido en tres roles principales que se tienen en cuenta a la hora de liderar un proyecto.

Propietario del producto (*Product Owner*)

El *Product Owner* se encarga de optimizar y maximizar el valor de un producto, donde él es la persona que se encarga de la gestión del flujo de valor del producto a través del *Product Backlog*. El *Product Owner* tienen otro rol importante como es el de representar al negocio o trabajo por lo cual aporta un valor al producto [18].

Maestro Scrum (*Scrum Master*)

El *Scrum Master* es el encargado de liderar al equipo para cumplir con los objetivos verificando que las reglas y procesos se cumplan, también ayuda a gestionar la reducción de impedimentos dentro del proyecto y trabaja a la par junto al *Product Owner* [19].

Equipo de desarrollo (*Development Team*)

El equipo de desarrollo es un equipo pequeño formado entre 3 a 9 personas, los cuales deben conocer específicamente el rol que debe cumplir cada uno, como equipo de desarrollo se auto-organizan y auto-gestionan para poder incrementar el software al final del ciclo de desarrollo [18]. Por último, los roles han sido divididos de la siguiente manera como se muestra en la

TABLA I.

TABLA I: Asignación de roles en el equipo de trabajo.

Nombre	ROL
Ing. Richard Rivera, PhD	Product Owner
Ing. Richard Rivera, PhD	Scrum Master
Edison Jumbo Jefferson Llumiquinga	Development Team

2.1.2 Artefactos

Las herramientas *Scrum* son las que garantizan la transparencia y el registro de información dentro de la metodología ágil, es decir son los recursos que reafirman la productividad y calidad del proyecto [20].

Historias de usuarios

Las historias de usuario son descripciones cortas y simples las cuales nos ayudan a describir una tarea concisa mediante la cual se recoge información sobre la funcionalidad con la que contara un software esto se da desde la perspectiva del usuario final o cliente [21]. La **TABLA II** detalla un ejemplo de las Historias de Usuario que se han elaborado mientras que el resto de Historias de Usuario necesarias para el desarrollo del sistema se encuentran detalladas en el Manual Técnico – Sección Historias de Usuario.

TABLA II: Historia de Usuario 1 – Iniciar sesión de usuario

Historia de Usuario	
Identificador (ID): HU001	Usuario: Cliente
Nombre Historia: Iniciar sesión del usuario	
Prioridad en negocio: Alta	Riesgo de desarrollo: Baja
Iteración Asignada: 4	
Responsable(es): Edison Jumbo-Jefferson Llumiquinga	
Descripción: El ingreso al sitio web se da a través de cualquier navegador mediante un login.	
Observación: Para poder visualizar las direcciones IPv4, realizar búsquedas filtradas y un análisis de puertos.	

Product Backlog

Es un inventario donde encontramos el trabajo a realizar en el producto final como los son: los requerimientos, casos de uso, tareas y dependencias, el *Product Backlog* es gestionado en exclusivo por el *Product Owner* donde tiene como prioridad los elementos de más valor en cada etapa y se detalla para el equipo de desarrollo [22]. La **TABLA III** nos detalla la elaboración del *Product Backlog*.

TABLA III: Elaboración de Product Backlog

Elaboración del Product Backlog

ID-HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD
HU001	Inicio de Sesión del usuario	4	Finalizado	Alta
HU002	Búsqueda de direcciones IPv4	1	Finalizado	Alta
HU003	Recolección de direcciones IPv4	2	Finalizado	Alta
HU004	Actualización de direcciones IPv4	1	Finalizado	Alta
HU005	Capturas de pantalla de puertos abiertos	1	Finalizado	Alta
HU006	Visualizar página informativa en el Sistema Web	4	Finalizado	Alta
HU007	Registro de usuarios	4	Finalizado	Alta
HU008	Editar Cuenta	4	Finalizado	Media
HU009	Iniciar y cerrar sesión de Administrador	3	Finalizado	Alta
HU010	Activación de cuenta	3	Finalizado	Alta
HU011	Visualizar direcciones IPv4 encontradas	4	Finalizado	Alta
HU012	Visualizar los detalles de cada dirección IPv4	4	Finalizado	Alta
HU013	Búsqueda y filtrado de direcciones IPv4	4	Finalizado	Alta
HU014	Visualizar los resultados de la búsqueda	4	Finalizado	Alta
HU015	Análisis de puertos por ciudad	4	Finalizado	Alta

Sprint Backlog

Un *Sprint* es considerado como un mini-proyecto el cual no puede durar más de un mes, cada sprint esta detallado de que es lo que se va a construir, un diseño y un plan flexible el cual permite que el producto final se lo desarrolle de forma rápida y eficiente [23]. Para el desarrollo del proyecto se establece 6 *Sprints* los cuales son: Configuración del

ambiente de desarrollo, Búsqueda de direcciones IPv4, Recolección de datos, Inicio de sesión de usuario Administrador, Inicio de sesión de usuario cliente, Pruebas de software, los cuales se lo detallan en el Manual Técnico – Sección *Sprint Backlog*.

2.2 Diseño de interfaces del sistema

Una vez establecidos los requerimientos necesarios se da paso a la creación de los bocetos (*Mockups*) para el Sistema Web.

2.2.1 Marvel

Es una herramienta que se utiliza para el diseño de prototipos, los diseños son funcionales e interactivos por ende al momento de mostrarlos al usuario, cliente o equipo de desarrollo se tiene una mejor comprensión del sistema que se va a realizar [24].

La **Fig. 1** ilustra el prototipo donde se muestran las direcciones Ipv4 recolectadas por el script desarrollado en Python además del buscador. Mientras que el resto de *Mockups* de la página web se encuentran detallados en el Manual Técnico - Sección Diseño de Interfaces.

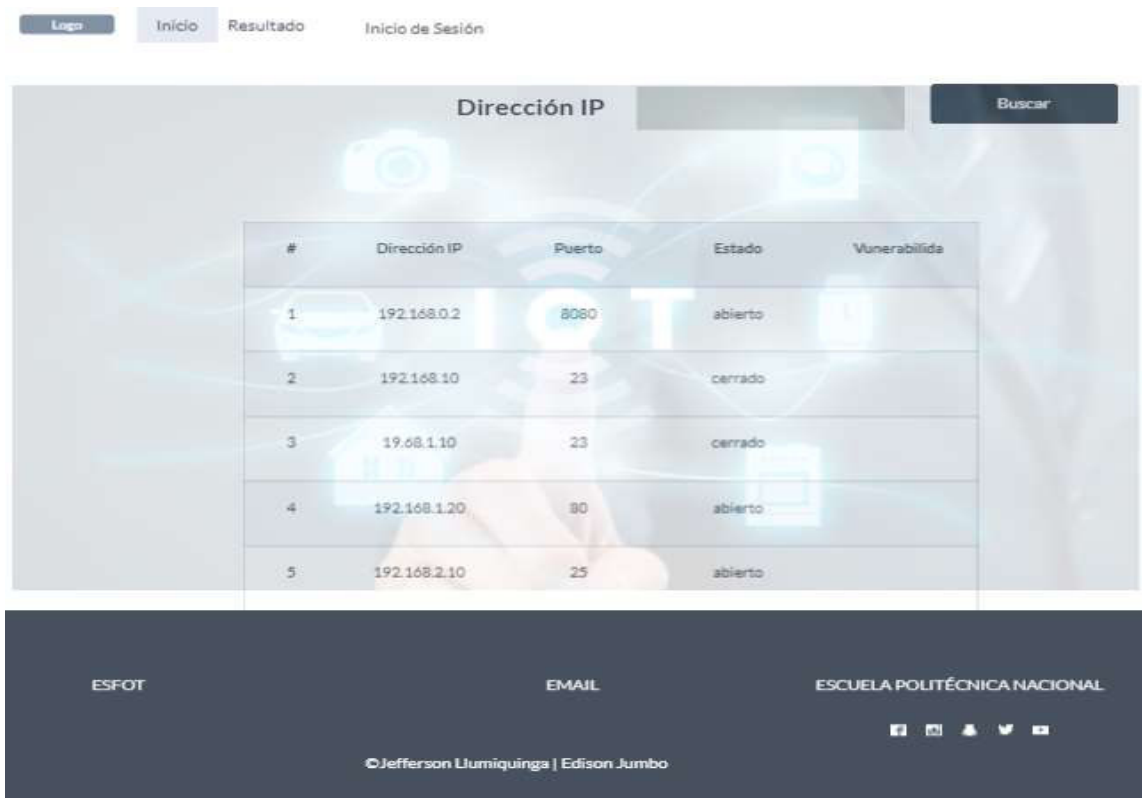


Fig. 1: Visualizar direcciones IPv4

2.3 Diseño de la arquitectura del sistema

Una vez que se determinan los roles y el diseño de interfaces se establece la arquitectura del sistema para el desarrollo del proyecto.

2.3.1 Patrón arquitectónico

Cumpliendo con las fases de la metodología y para poder cumplir tanto con la codificación de la herramienta y el sistema propuesto se ha utilizado el patrón MODELO, VISTA, CONTROLADOR (MVC) el cual es una arquitectura de software donde separa los datos de una aplicación, interfaz de usuario y la lógica en 3 componentes distintos [25].

- **Modelo:** Representación de los datos manejados en el sistema.
- **Vista:** Interfaz gráfica para la interacción con el cliente.
- **Controlados:** Se encarga de manejar las peticiones que el usuario realiza.

2.3.2 Sistema Web

La Fig. 2 muestra el patrón de arquitectura que se implementa para el desarrollo de la herramienta y el sistema web esto en función a las herramientas utilizadas.

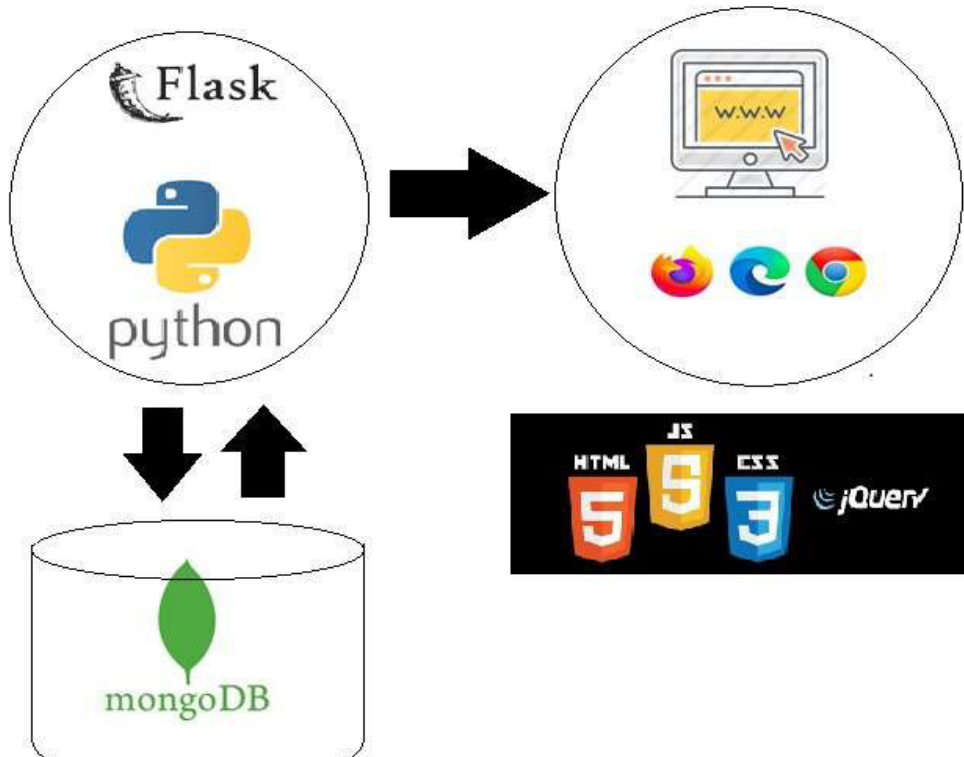


Fig. 2: Patrón Arquitectónico

2.4 Herramientas de desarrollo

Establecer las herramientas correctas, permite alinear los objetivos del proyecto en su conjunto con el equipo de trabajo, procesos, tareas, costes, etc.

A continuación, la **TABLA IV** describe las herramientas que se han utilizado en el presente proyecto.

TABLA IV: Herramientas para el desarrollo del Sistema Web y Script

Herramienta	Justificación
Flask (back-end)	<i>Flask</i> es un Microframework escrito en <i>Python</i> mediante el cual se pueden crear Aplicaciones Web bajo el patrón MVC, <i>Flask</i> además cuenta con un gran conjunto de plugins que son fáciles de instalar y nos ayudan al desarrollo de una Aplicación Web [26].
Python	Lenguaje de programación que se destaca por trabajar en diversas plataformas y con grandes volúmenes de datos, ya que, al ser multiplataforma, favorece su extracción y procesamiento [27].
MongoDB	Sistema de base de datos no relacional de código abierto, orientado a documentos por medio de una estructura BSON. Permite trabajar de manera distribuida utilizando el modelo de maestro esclavo [28].
Github	Es una plataforma de trabajo colaborativa en línea, que se apoya sobre el sistema de control de versiones Git. Permite operar sobre el código fuente de los programas y llevar un desarrollo ordenado [29].

2.4.1 Librerías

La **TABLA V** muestra las librerías que se han utilizado para el desarrollo del presente proyecto.

TABLA V: Librerías utilizadas para el desarrollo del Sistema Web y Script

Herramienta	Justificación
PyMongo (data-base)	Controlador oficial de MongoDB para <i>Python</i> que facilita la conectividad y manipulación de datos [30].
Ice-cream (debug)	Permite que la impresión del terminal sea más legible al mostrar parte del código con su respectiva salida. Eliminando las típicas Depuraciones Print() o Log() [31].
Logging(Logs)	Herramienta útil para prevenir errores y llevar a cabo un análisis por medio de mensajes de registro de diferentes niveles tanto en los archivos como en la consola [32].
Socket	Proveen de un canal de comunicación bidireccional entre un programa del cliente y un programa del servidor en una red local o remota [33].
DNS	Permite realizar consultas y manipular zonas, mensajes, nombres y registros de DNS [34].
Selenium	Permite realizar controles de calidad para fines de pruebas de automatización y también se pueden usar para tomar capturas de pantalla automáticamente [35].
Pygeoip	Provee con una gran potencia para filtrar datos referentes al geoposicionamiento de una dirección IPv4 por medio de bases de datos GeolP binarios [36].
ipWhois	Permite conocer con exactitud la información técnica y los datos de registro de la persona o empresa que registro el dominio [37].

3. RESULTADOS Y DISCUSION

En esta sección se muestra de manera detallada la implementación y resultados que han sido obtenidos en cada una de las tareas de los *Sprint* de desarrollo establecidos previamente, al igual que los resultados obtenidos en cada una de las pruebas realizadas tanto en el *Script* como en el sistema web.

3.1 Sprint 0. Configuración del ambiente de desarrollo

Entre los parámetros establecidos dentro del *Sprint Backlog* que se detalla en el Manual Técnico – Sección *Sprint Backlog*. El *Sprint 0* se compone de varias tareas indispensables para obtener un ambiente de desarrollo optimo que nos permita ejecutar el *Product Backlog*. A continuación, describimos las tareas que se llevan a cabo dentro del *Sprint*.

- Elaboración de la base de datos en MongoDB.
- Estructura del proyecto.
- Roles de Usuarios.
- Requerimientos para el script de búsqueda y almacenamiento de direcciones IPv4.
- Requerimientos del Sistema Web.

3.1.1 Elaboración de la Base de datos en MongoDB

Para la elaboración de la base de datos para el sistema se utiliza la base de datos NoSQL MongoDB que es un sistema orientado a documentos y código abierto, MongoDB ofrece una gran escalabilidad, flexibilidad y un modelo para consultas [38].

En la **Fig. 3** ilustramos las colecciones que conforman la base de datos NoSQL, mientras que en la **Fig. 4** los documentos almacenados al momento que el *Script* recolecta las direcciones IPv4.

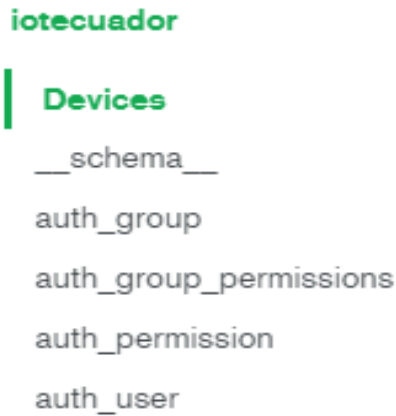


Fig. 3: Colección en MongoDB

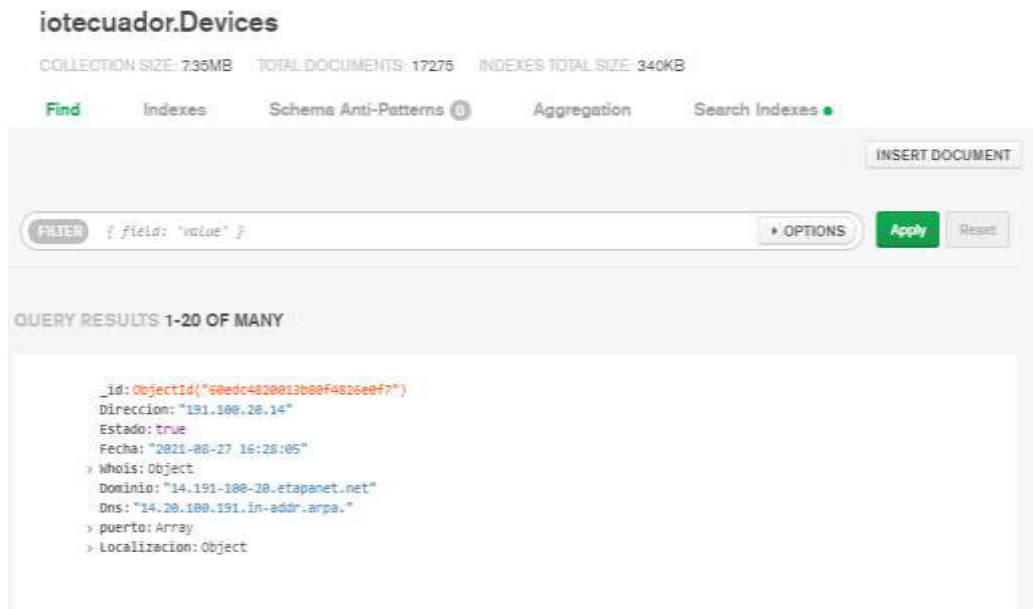


Fig. 4: Direcciones IPv4 almacenadas

3.1.2 Estructura del proyecto

Se utiliza Visual Studio Code como herramienta para crear tanto el *Script* como el Sistema Web tomando en cuenta el patrón arquitectónico MVC. La Fig. 5 muestra los directorios necesarios para crear el *Script* mientras que la Fig. 6 muestra los directorios para crear el Sistema Web.

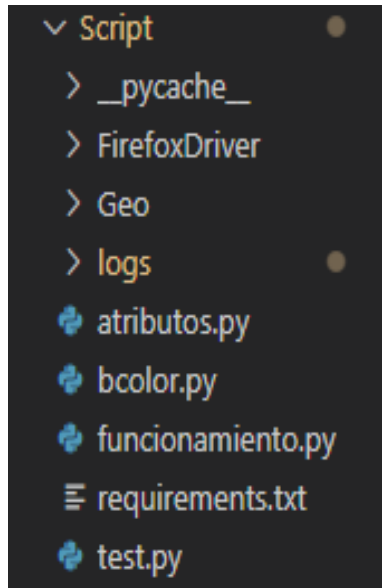


Fig. 5: Script de búsqueda y recolección de direcciones IPv4

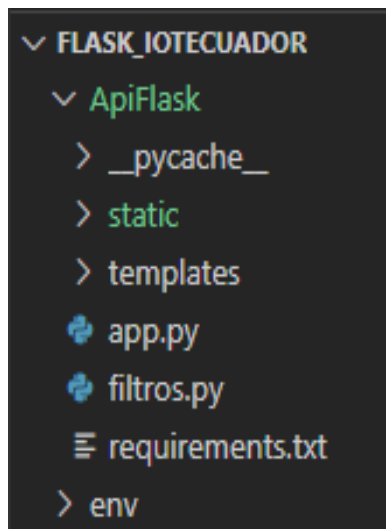


Fig. 6: Estructura del Sistema Web

3.1.3 Roles del usuario

La **Fig. 7** y **Fig. 8** muestran el manejo del administrador tanto en el *Script* como en el Sistema Web mientras que en la **Fig. 9** se muestra el manejo del usuario y su finalidad en el Sistema Web.

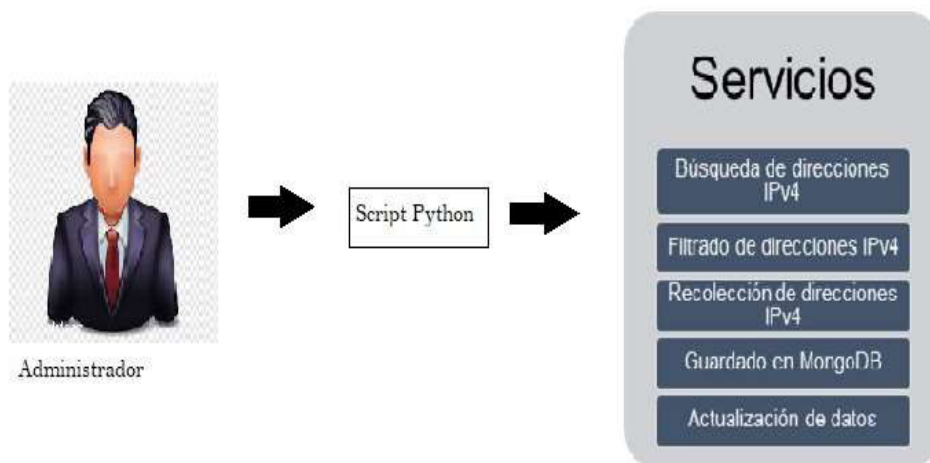


Fig. 7: Script Python

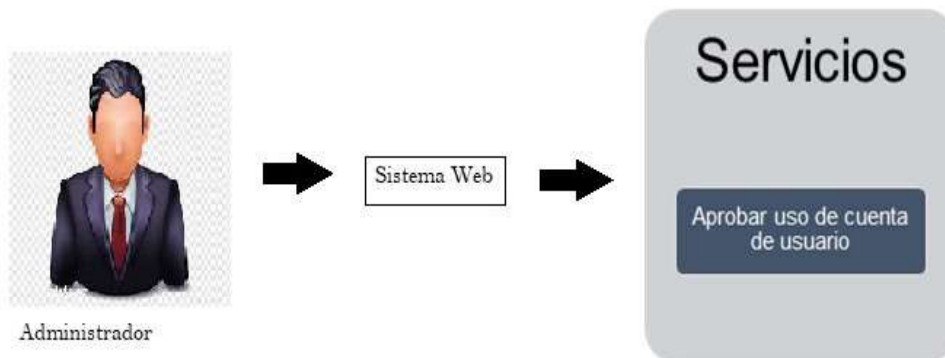


Fig. 8: Activación de cuentas de usuarios

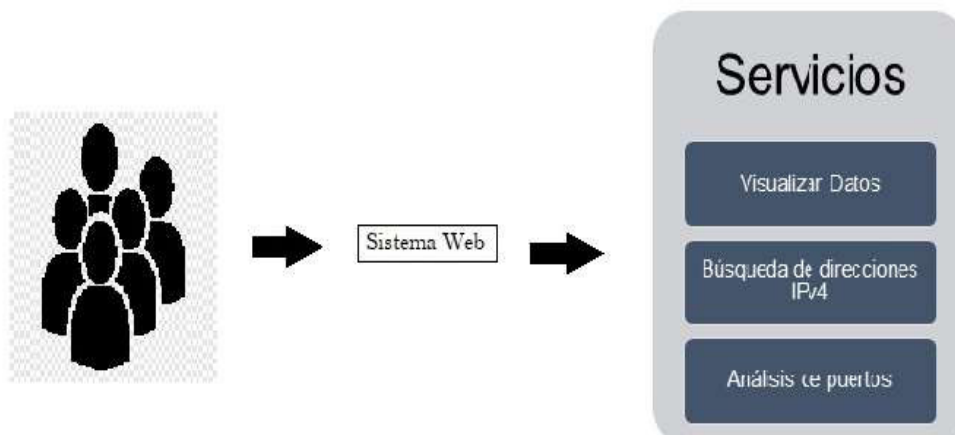


Fig. 9: Sistema Web

3.1.4 Requerimientos para el *Script* de búsqueda y almacenamiento de direcciones IPv4

Búsqueda de direcciones IPv4

Únicamente el usuario administrador corre el *Script* desarrollado en *Python* para empezar la búsqueda de direcciones IPv4 en territorio ecuatoriano.

Filtrado de direcciones IPv4

El usuario administrador genera direcciones IPv4 que son filtradas y se deben encontrar dentro del archivo .dat de geolocalización.

Recolección

En el momento que el usuario administrador corre el *Script* de búsqueda de direcciones IPv4 al mismo instante las direcciones encontradas son almacenadas en una colección dentro de MongoDB.

Actualización de puertos

Si las direcciones IPv4 almacenadas en MongoDB tienen una inactividad mayor a los 30 días se procede a la actualización, es decir, la búsqueda de nuevos puertos y capturas si se da el caso.

Capturas

El usuario administrador determina que se van a realizar capturas de pantalla en todos los puertos abiertos que lleguen a tener cada dirección IPv4, esta función se la da con ayuda de un archivo ejecutable que ocupa Google Chrome para realizar dichas capturas de cada puerto.

3.1.5 Requerimientos para el Sistema Web

Autenticación de usuarios

Una vez que el administrador haya activado la cuenta del usuario, el antes mencionado ya tiene la posibilidad de acceder al Sistema Web mediante correo y contraseña para tener el permiso de uso en el filtrado y búsqueda de direcciones IPv4.

Visualización y Búsqueda

El usuario final puede observar las direcciones IPv4, los puertos recolectados por el *Script* y realizar una búsqueda filtrada por medio de los siguientes parámetros.

- Dirección.
- Puerto.
- Geolocalización.

3.2 Sprint 1. Buscar direcciones IPv4

Partiendo de la planificación realizada en el *Sprint Backlog* que se encuentra detallado en el Manual Técnico – *Sprint Backlog*.

El *Sprint 3* se compone de las siguientes actividades indispensables para la búsqueda de direcciones IPv4.

A continuación, se muestran las tareas determinadas dentro del Sprint:

- Buscar direcciones IPv4
- Actualizar direcciones IPv4
- Capturar pantalla de puertos abiertos

3.2.1 Buscar direcciones IPv4

En la **Fig. 10** el administrador observa un menú mediante el cual encontramos 3 opciones las cuales son la búsqueda de direcciones IPv4, información acerca de la herramienta y la opción de salir, a continuación, en la **Fig. 11** se observa la orden de cuantas direcciones IPv4 deseamos buscar.



Fig. 10: Menú de opciones

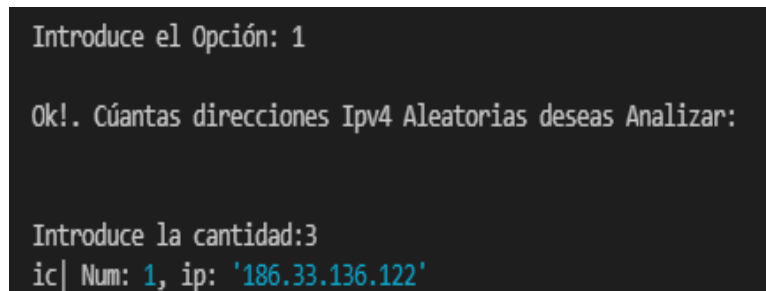


Fig. 11: Cantidad de direcciones IPv4 a buscar

3.2.2 Actualizar direcciones IPv4

En la Fig. 12 se muestra el proceso mediante el cual se lleva a cabo la actualización de dirección IPv4 si se llega a cumplir las condiciones dadas en el *Script*, como lo son: lapso mayor a 30 días en las direcciones con estado True y un lapso de 15 días para las direcciones con estado False.

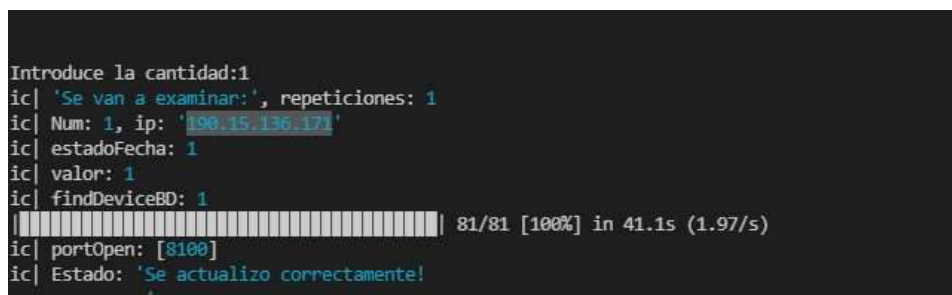
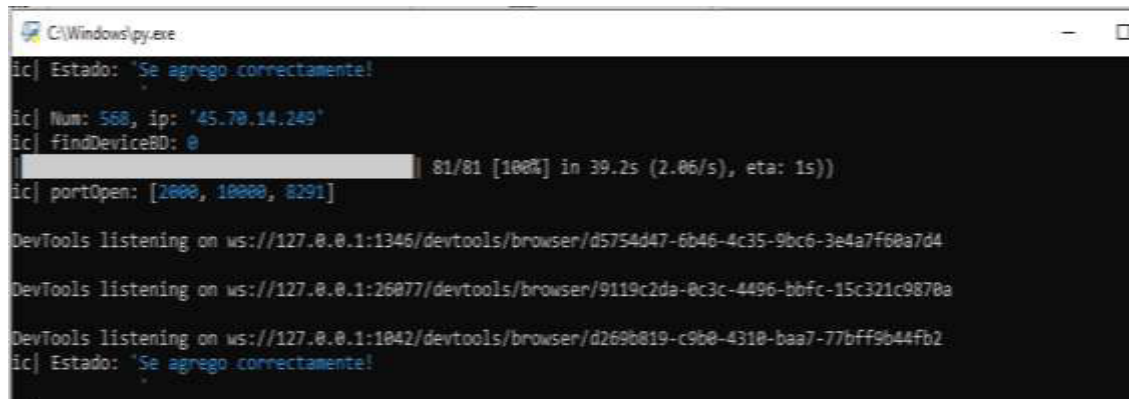


Fig. 12: Actualización de direcciones IPv4

3.2.3 Capturar pantalla de puertos abiertos

Como se muestra en la Fig. 13 cuando una dirección IPv4 tiene N número de puertos abiertos para cada uno de ellos se le realiza una captura de pantalla.



```
C:\Windows\py.exe
ic| Estado: "Se agrego correctamente!"
ic| Num: 568, ip: '45.78.14.249'
ic| findDeviceBD: @
[redacted] [ 81/81 [100%] in 39.2s (2.06/s), eta: 1s))
ic| portOpen: [2000, 10000, 8291]
DevTools listening on ws://127.0.0.1:1346/devtools/browser/d5754d47-6b46-4c35-9bc6-3e4a7f60a7d4
DevTools listening on ws://127.0.0.1:26077/devtools/browser/9119c2da-0c3c-4496-bbfc-15c321c9870a
DevTools listening on ws://127.0.0.1:1042/devtools/browser/d269b819-c9b0-4310-baa7-77bfff9b44fb2
ic| Estado: "Se agrego correctamente!"
```

Fig. 13: Captura de pantalla

3.3 Sprint 2. Recolección de datos

Partiendo de la planificación realizada en el *Sprint Backlog* que se encuentra detallado en el Manual Técnico - Sección *Sprint Backlog*.

En el *Sprint 2* se detalla la funcionalidad de como el *Script* recolecta las direcciones IPv4 encontradas.

A continuación, se muestra la tarea determinada dentro del Script

- Recolectar direcciones IPv4

3.3.1 Recolectar direcciones IPv4

En la Fig. 14 se observa el mensaje que nos brinda el *Script* al momento de encontrar una dirección IPv4, se analizó si se encuentran puertos abiertos, se analizó si hay la posibilidad de tomar una captura de pantalla de un puerto abierto en específico y por último en la Fig. 15 se observa que la dirección IPv4 ha sido guardada con éxito en MongoDB.


```
ic| 'Se van a examinar:', repeticiones: 500
ic| Num: 1, ip: '186.70.24.98'
ic| findDeviceBD: 0
| _____ | 81/81 [100%] in 40.7s (1.99/s), eta: 0s))
ic| portsNumbers: 0
ic| Estado: 'Se agrego correctamente!'
ic| Num: 2, ip: '190.214.8.196'
ic| findDeviceBD: 0
| _____ | 81/81 [100%] in 40.7s (1.99/s), eta: 0s))
ic| portsNumbers: 0
ic| Estado: 'Se agrego correctamente!'
ic|
```

Fig. 14: Recolección de direcciones IPv4

```
FILTER: {{Direccion: "190.214.8.196"}} OPTIONS Apply
PROJECT: { field: 0 }
SORT: { field: -1 }
COLLATION: { locale: 'simple' }

QUERY RESULTS 1-1 OF 1

  _id: ObjectId("614e32c445b9968a0529da08")
  Direccion: "190.214.8.196"
  Estado: false
  Fecha: "2021-09-24 15:19:14"
  > Localizacion: Object
  Whois: null
  Dominio: null
  Dns: null
  puerto: null
```

Fig. 15: Recolección de datos en MongoDB

En la Fig. 16 representamos los índices para mejorar el rendimiento de búsqueda al momento de utilizar el Sistema Web.

Name, Definition, and Type	Size	Usage	Properties	Action
id <small>REGULAR</small>	488.0KB	< 1/min since Wed Sep 23 2023		
TPv4 <small>REGULAR</small>	456.0KB	< 1/min since Wed Sep 23 2023	<small>UNIQUE</small>	Drop Index
'Geolocalizacion' <small>REGULAR</small>	192.0KB	< 1/min since Wed Sep 23 2023		Drop Index
'Puentes' <small>REGULAR</small>	144.0KB	< 1/min since Wed Sep 23 2023		Drop Index

Fig. 16: Índices del documento Devices

3.4 Sprint 3. Inicio de sesión del usuario administrador

Partiendo de la planificación realizada en el *Sprint Backlog* que se encuentra detallado en el Manual Técnico – Sección *Sprint Backlog*.

El *Sprint 3* se compone de las siguientes actividades indispensables para el ingreso del usuario administrador en el Sistema Web.

A continuación, se muestra las tareas determinadas dentro del Sprint:

- Iniciar y cerrar sesión de Administrador
- Activar cuentas

3.4.1 Iniciar y cerrar sesión de Administrador

La Fig. 17 muestra los campos del formulario de login y las validaciones respectivas previo al ingreso al sistema, se tiene en cuenta que las credenciales del usuario administrador son proporcionadas por el equipo de desarrollo. A continuación, el sistema verifica el rol del usuario al que pertenece y lo redirecciona al sitio donde se hace posible activar o no activar las cuentas del resto de usuarios como se muestra en la Fig. 18.

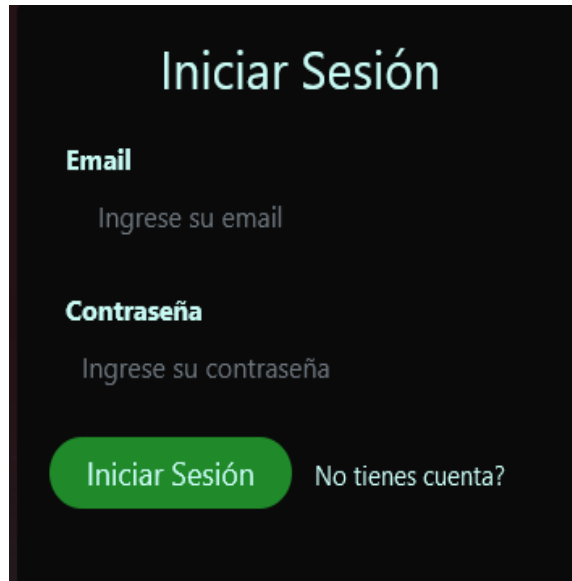


Fig. 17: Inicio de sesión rol administrador



Fig. 18: Panel para activar cuentas

3.5 Sprint 4. Inicio de sesión del usuario cliente

Partiendo de la planificación realizada en el *Sprint Backlog* que se encuentra detallado en el Manual Técnico-Sección *Sprint Backlog*.

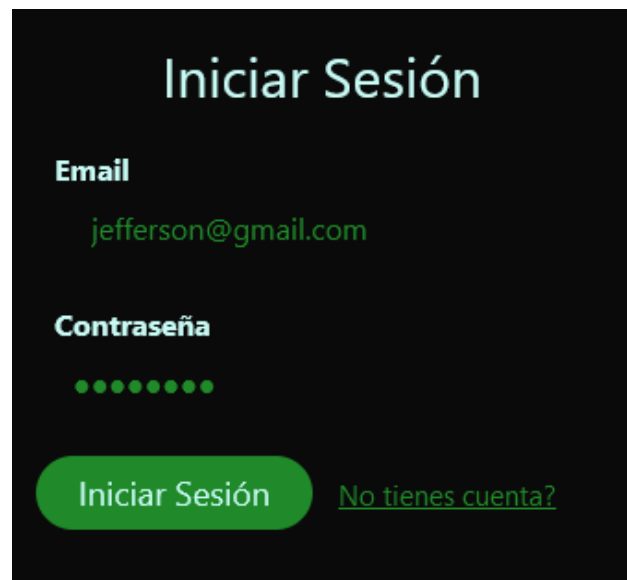
En el *Sprint 4* se detalla la funcionalidad que tiene el sistema web.

A continuación, se muestra las tareas determinadas dentro del *Sprint*:

- Inicio de sesión con rol usuario.
- Registro de usuario cliente.
- Editar perfil.
- Visualización de página de inicio.
- Visualizar direcciones IPv4.
- Buscar direcciones IPv4.
- Análisis de direcciones y puertos encontrados.

3.5.1 Inicio de sesión con rol usuario

La **Fig. 19** muestra los campos de formulario de login y las validaciones respectivas previo al ingreso del sistema cabe recalcar que para ambos roles el proceso es igual, cuando el sistema verifica el rol del usuario nos redirige hacia el *Dashboard* como se muestra en la **Fig. 20**.



Iniciar Sesión

Email
jefferson@gmail.com

Contraseña
●●●●●●●

Iniciar Sesión [No tienes cuenta?](#)

Fig. 19: Inicio de sesión rol usuario



Fig. 20: Información en el Dashboard

3.5.2 Registro de usuario cliente

La Fig. 21 detalla el formulario y la validación de los campos para registrar la información del cliente, cabe recalcar que al momento de registro el usuario cliente debe esperar que el administrador active su cuenta para poder ingresar al *Dashboard*.

The registration form includes the following fields and instructions:

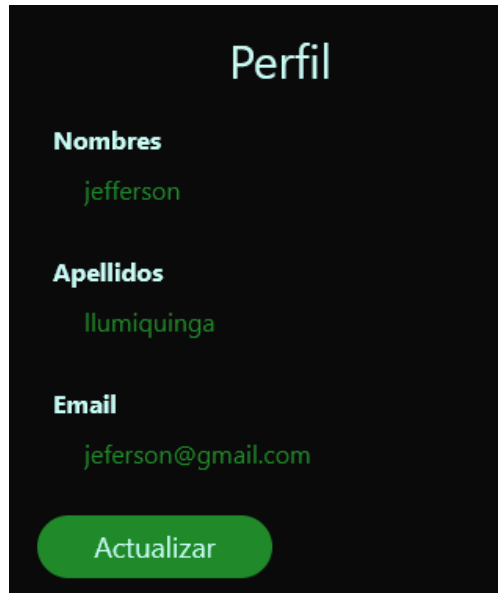
- Nombres:** Ingrese sus Nombres
- Apellidos:** Ingrese sus Apellidos
- Email:** Ingrese un email
- Pregunta:** ¿Por qué desea usar la aplicación?
- Contraseña:** Debe tener al menos (1-9, # \$ @, A-Z)
- Repita su contraseña:** Repita la contraseña

Buttons: **Regístrate** and [Tienes una cuenta?](#)

Fig. 21: Registro de usuario final

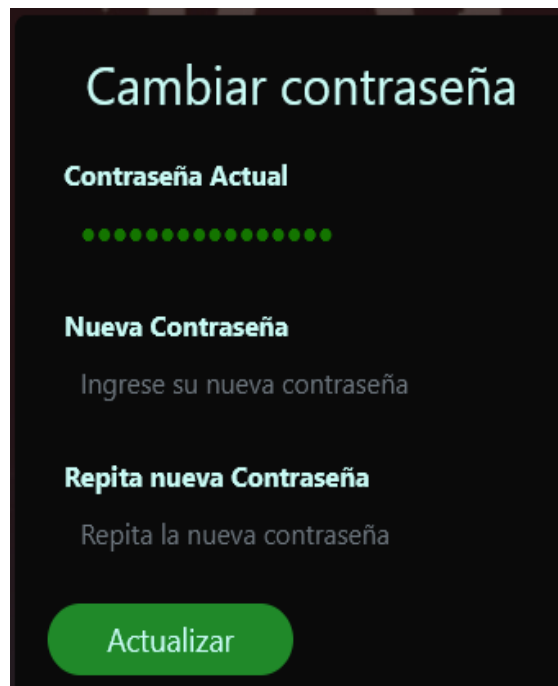
3.5.3 Editar Perfil

La **Fig. 22** muestra la configuración del usuario final en donde a través de un formulario se puede editar la información personal y adicional su correo en la **Fig. 23** se tiene un formulario donde el usuario puede cambiar su contraseña, para realizar esta función primero el usuario coloca su contraseña actual y a continuación la nueva contraseña.



The screenshot shows a dark-themed form titled "Perfil". It contains three sections: "Nombres" with the value "jefferson", "Apellidos" with the value "llumiquinga", and "Email" with the value "jeferson@gmail.com". At the bottom, there is a green rounded button labeled "Actualizar".

Fig. 22: Editar Perfil



The screenshot shows a dark-themed form titled "Cambiar contraseña". It contains three sections: "Contraseña Actual" with a masked input field of 12 green dots, "Nueva Contraseña" with a placeholder "Ingrese su nueva contraseña", and "Repita nueva Contraseña" with a placeholder "Repita la nueva contraseña". At the bottom, there is a green rounded button labeled "Actualizar".

Fig. 23: Editar contraseña

3.5.4 Visualización de página de inicio

La primera información que se encuentra en la página de inicio es un mapa que muestra la cantidad de puertos por provincia como se muestra en la Fig. 24.

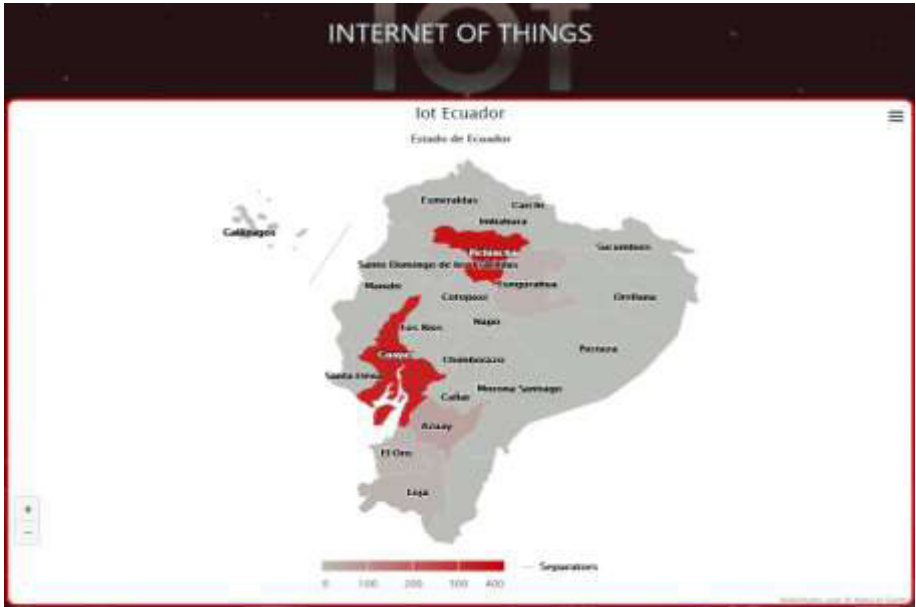


Fig. 24: Mapa cantidad de puertos por provincia

En la Fig. 25 se muestra la información acerca de productos conectados a una red wifi e información acerca de los desarrolladores.



Fig. 25: Aparatos IoT e información de desarrolladores

3.5.5 Visualizar direcciones IPv4

En el *Dashboard* observamos el total de direcciones IPv4 que se ha recolectado por MongoDB y se lo muestra en una tabla como se observa en la **Fig. 26**.



LISTA DE DIRECCIONES IPv4
Se ha encontrado : 936 Direcciones IPv4

Dirección: ¿Qué deseas buscar? Search

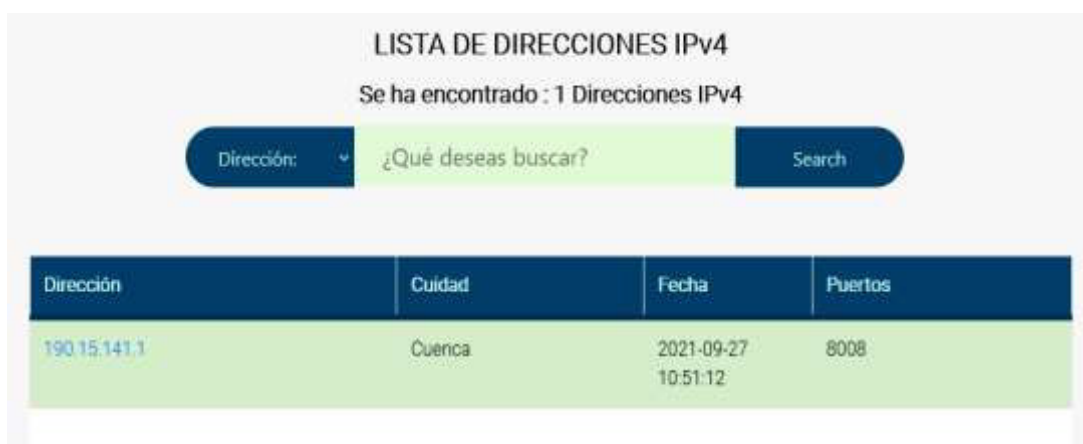
Dirección	Ciudad	Fecha	Puertos
192.141.38.50	None	2021-09-24 17:40:38	80 2000
185.4.179.90	Quito	2021-09-24 15:40:38	443 9000
190.155.84.187	Guayaquil	2021-09-24 11:36:21	8080
45.6.227.129	None	2021-09-24 11:30:07	53
209.7.200.129	Quito	2021-09-24	5150 5160

Fig. 26: Visualización de direcciones IPv4

3.5.6 Buscar direcciones IPv4

En la **Fig. 27**, **Fig. 28** y **Fig. 29** se observa el motor de búsqueda que se ha implementado en la misma página donde podemos visualizar las direcciones IPv4, este motor de búsqueda cuenta con 3 opciones como lo son:

- Dirección.
- Puerto.
- Ciudad.



LISTA DE DIRECCIONES IPv4
Se ha encontrado : 1 Direcciones IPv4

Dirección: ¿Qué deseas buscar? Search

Dirección	Ciudad	Fecha	Puertos
190.15.141.1	Cuenca	2021-09-27 10:51:12	8008

Fig. 27: Búsqueda por la opción de dirección

LISTA DE DIRECCIONES IPv4
Se ha encontrado : 263 Direcciones IPv4

Dirección: Search

Dirección	Ciudad	Fecha	Puertos
200.7.195.124	Guayaquil	2021-07-13 12:35:58	22 23 80 443
190.15.136.171	None	2021-08-27 16:00:51	22 23 53 80 81 110 180 443 873 2323 5000 5001 5094 5150 5160 8080 8100 8443 8883 49152 52869 56000 1728 3001 8008 8009 10001 223 1080 1935 2332 8888 9100 2000 2001 2002 2003 2004 2005 2006 2007 21 554 888 1159 1160 1161 1435 1518 3389 4550 5005 5400 5550 6550 7000 8000 8081 8090 8150 8866 9000 9650 9999 10000 18004 25001 30001 34567 37777 69 161 162 4786 5431 8291 37215 53413
201.234.193.196	Quito	2021-09-16 15:45:52	80 8080 8443 554 8000
190.95.145.169	Guayaquil	2021-07-13 14:52:37	22 80 7547
181.196.245.50	Machala	2021-07-13	23 80 8291

Fig. 28: Búsqueda por la opción de puerto

LISTA DE DIRECCIONES IPv4
Se ha encontrado : 18 Direcciones IPv4

Ciudad: Search

Dirección	Ciudad	Fecha	Puertos
200.0.30.62	Loja	2021-07-13 18:03:20	8008 2000
200.0.31.251	Loja	2021-07-19 19:05:46	8008 2000
186.68.9.56	Loja	2021-07-19 21:56:20	7547
192.188.50.216	Loja	2021-07-20 13:44:43	8008 2000
186.68.11.28	Loja	2021-07-26 18:32:59	7547
200.0.30.245	Loja	2021-07-27 14:50:26	8008 2000

Fig. 29: Búsqueda por la opción de ciudad

3.5.7 Análisis de direcciones y puertos encontrados

En la Fig. 30 se observa un análisis de puertos al momento de la búsqueda por ciudad donde se muestra por resultado una gráfica que nos indica la cantidad de puertos activos en la ciudad buscada.



Fig. 30: Análisis de puertos por ciudad

3.6 Sprint 5. Pruebas de software

Partiendo de los parámetros establecidos en el *Sprint Backlog* detallado en el Manual Técnico – Sección *Sprint Backlog*.

El *Sprint 5* se compone de la prueba de software para verificar el correcto funcionamiento tanto del Script como del Sistema Web.

3.6.1 Pruebas funcionales

Las pruebas funcionales son definidas como una fuente de requerimiento para comprobar el sistema o herramienta desarrollada para verificar su correcto funcionamiento y saber si cumple con los requisitos del usuario final [39].

3.6.1.1 Pruebas de Aceptación

Las pruebas de aceptación son pruebas formales que se realizan para verificar si los requerimientos están siendo cumplidos esto debido a las necesidades satisfaciendo las necesidades del usuario final [40].

La **TABLA VI** detalla un ejemplo de las Pruebas de aceptación que se han elaborado mientras que el resto de Pruebas de aceptación necesarias para comprobar el funcionamiento del sistema se encuentran detalladas en el Manual Técnico – Sección Pruebas de aceptación.

TABLA VI: Pruebas de aceptación - Ingreso al Sistema Web

Prueba de Aceptación
Identificador (ID): PA001
Identificador historia de Usuario: HU001
Nombre para prueba de aceptación: Ingreso al Sistema Web usuario cliente.
Descripción: El usuario ingresa al Sistema Web mediante su correo y contraseña.
Pasos de ejecución: Dirigirse al Sistema Web en cualquier navegador. Escribir correo y contraseña. Clic en iniciar sesión.
Resultado deseado: El Sistema Web permite ingresar al usuario con sus credenciales.
Evaluación de la prueba: Se comprueba el resultado esperado. Aprobación del cliente 100%.

3.6.1.2 Pruebas de Compatibilidad

Las pruebas de compatibilidad son de suma importancia para poder mostrar la calidad adecuada de un software y verificar su funcionalidad en todos los navegadores [41].

Sistema Web

Estas pruebas han sido realizadas en 3 de los navegadores más populares e implementados en la actualidad como se muestra en la **TABLA VII**. Los resultados obtenidos tras la ejecución de esta prueba se detallan en el Manual Técnico – Sección Pruebas de compatibilidad

TABLA VII: Navegadores utilizados para la ejecución de la prueba

Nombre	Versión	Observación
Google Chrome	93.0.4577.82	Totalmente funcional.
Mozilla Firefox	92.0	Parcialmente Funcional.
Microsoft Edge	93.0.961.52	Totalmente funcional.

3.7 Discusión

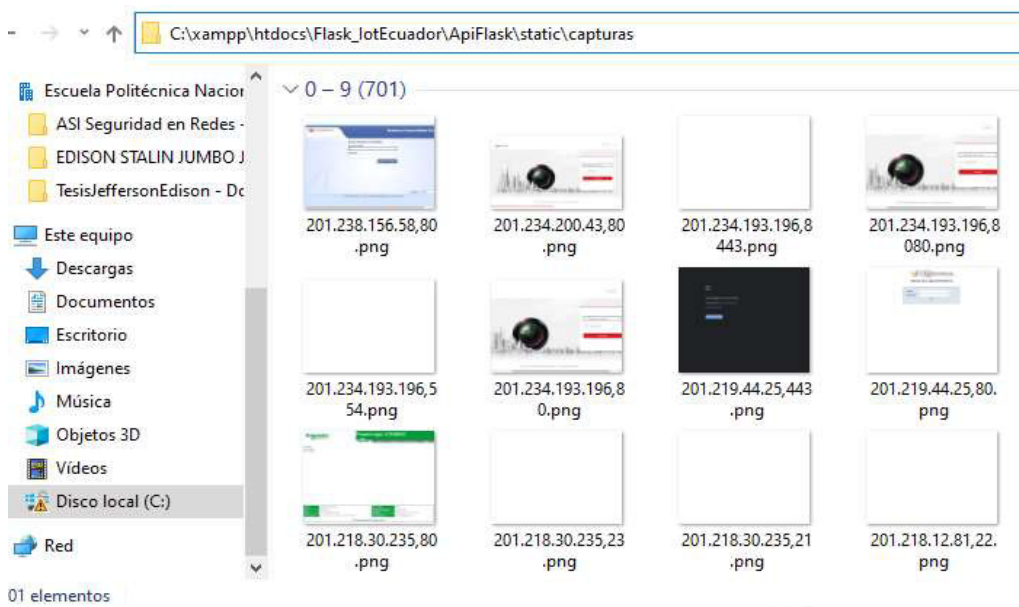
Después de recolectar en MongoDB hasta el día 26 de septiembre 2021 un total de 23250 direcciones IPv4 alrededor del Ecuador, en nuestro Sistema Web únicamente se muestran un total de 936 direcciones IPv4 las cuales cuentan con algún tipo de puerto activo y al cual se le hace posible el acceso tal como vemos en la **Fig. 31** y **Fig. 32**

Fig. 31: Cantidad de direcciones IPv4 en MongoDB

LISTA DE DIRECCIONES IPv4
Se ha encontrado : 936 Direcciones IPv4

Fig. 32: Cantidad de datos con puertos abiertos

Como una de las funciones de la herramienta se tiene la funcionalidad de la captura de pantalla de puertos abiertos, en total hasta el día de hoy tenemos un total de 701 capturas guardadas en nuestro directorio capturas como se muestra en la **Fig. 33**.

**Fig. 33:** Total de capturas en el directorio

De las 701 capturas que se tiene se observa que el puerto 80 es el que cuenta con el mayor número de capturas por ser un puerto muy común dentro de las direcciones IPv4 encontradas.

En la **Fig. 34** se muestra la cantidad de direcciones IPv4 que contienen el puerto 80 que al ser un puerto muy común podemos observar que tiene un número significativo de direcciones IPv4 con dicho puerto **Fig. 35**.

LISTA DE DIRECCIONES IPv4

Se ha encontrado : 261 Direcciones IPv4

Dirección:

Dirección	Ciudad	Fecha	Puertos
200.7.195.124	Guayaquil	2021-07-13 12:35:58	22 23 80 443

Fig. 34: Numero de direcciones IPv4 con puerto 80

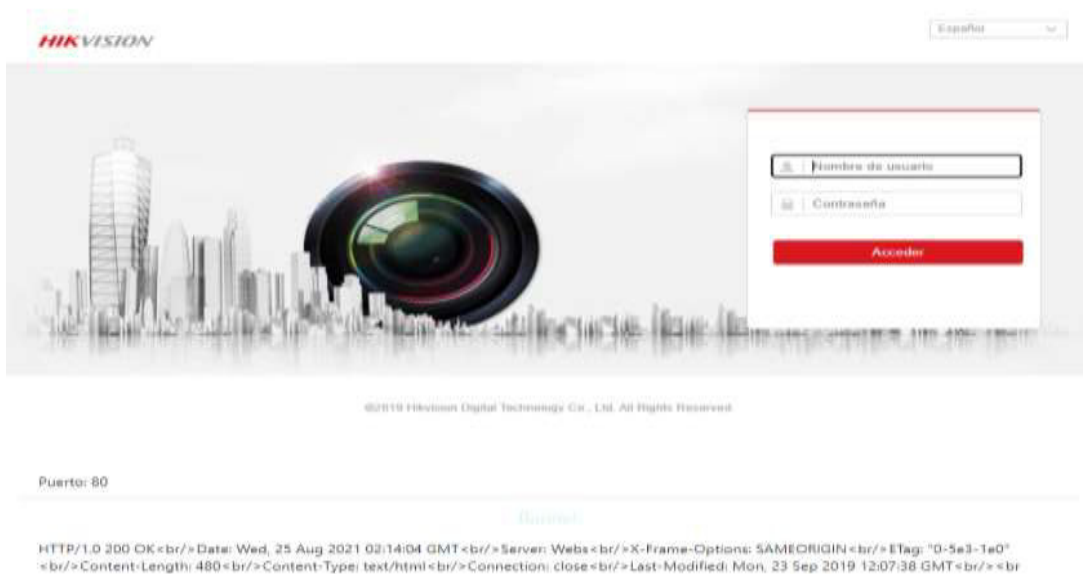


Fig. 35: Captura de pantalla en puerto 80

Se realiza un estudio en las dos provincias más grandes del Ecuador como lo son Pichincha y Guayas donde se encuentra el mayor número direcciones IPv4 con puertos abiertos. Pichincha tiene un total de 382 direcciones IPv4, donde se tiene la cantidad de 143 direcciones IPv4 con puerto 80 como se puede ver en la **Fig. 36** y 83 direcciones IPv4 con puerto 443 como se puede ver en la **Fig. 37** siendo las 2 más relevantes en la provincia.

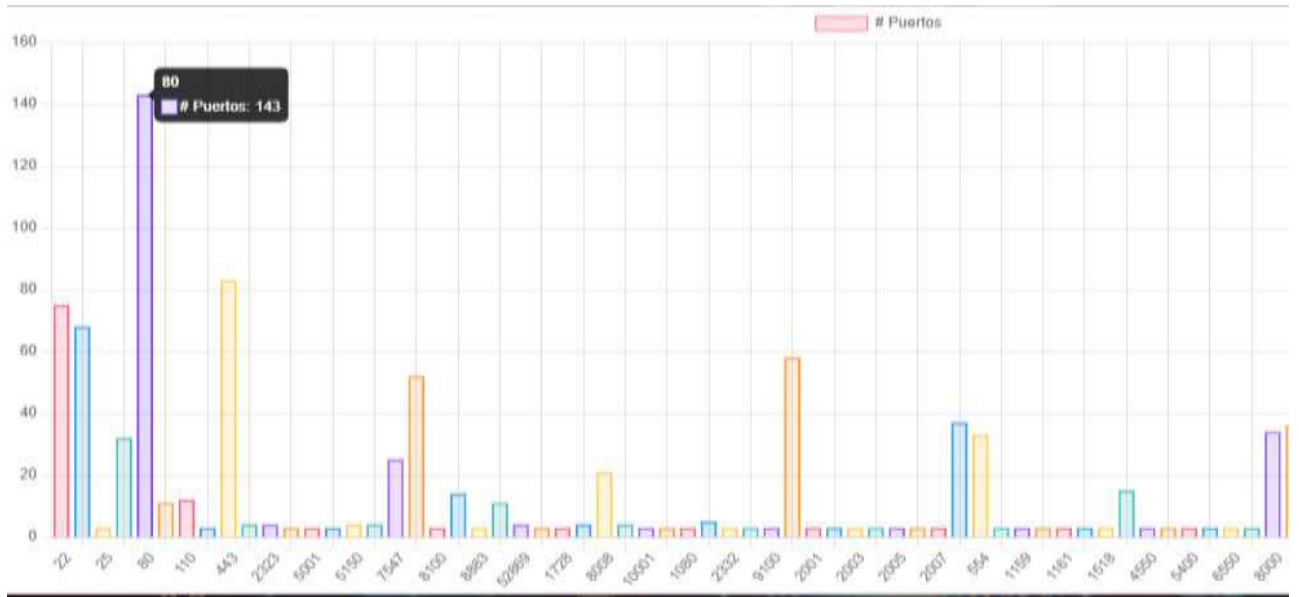


Fig. 36: Puerto 80 en Pichincha

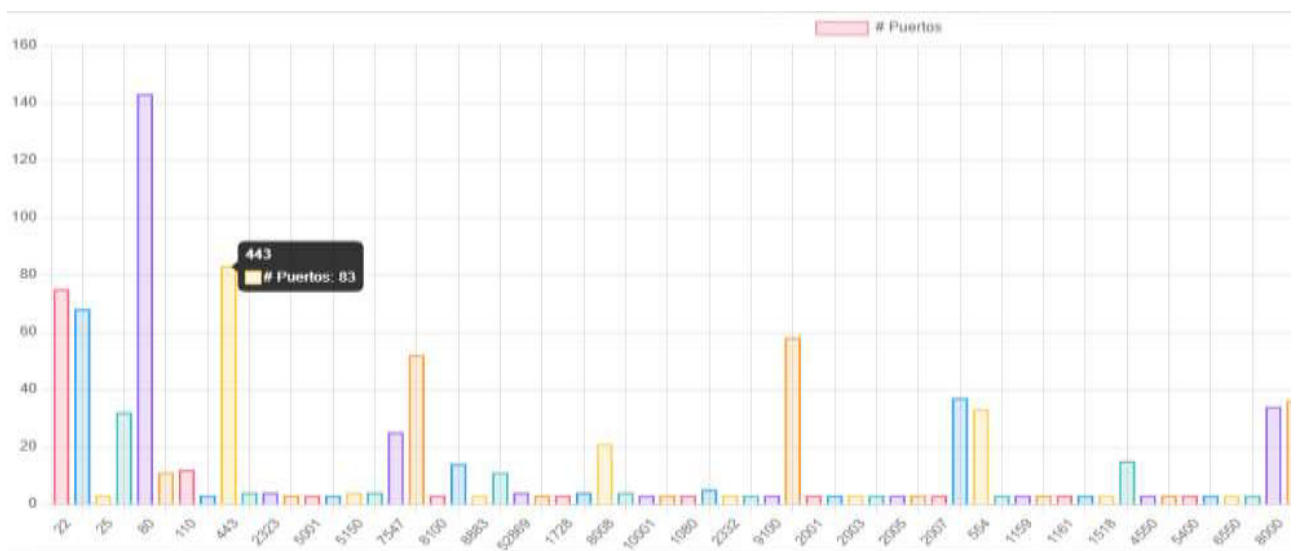


Fig. 37: Puerto 443 en Pichincha

Guayas tiene un total de 348 direcciones IPv4, donde se tiene la cantidad de 95 direcciones IPv4 con puerto 22 como se puede ver en la Fig. 38 y 83 direcciones IPv4 con puerto 80 como se puede ver en la Fig. 39 siendo las 2 más relevantes en la provincia.

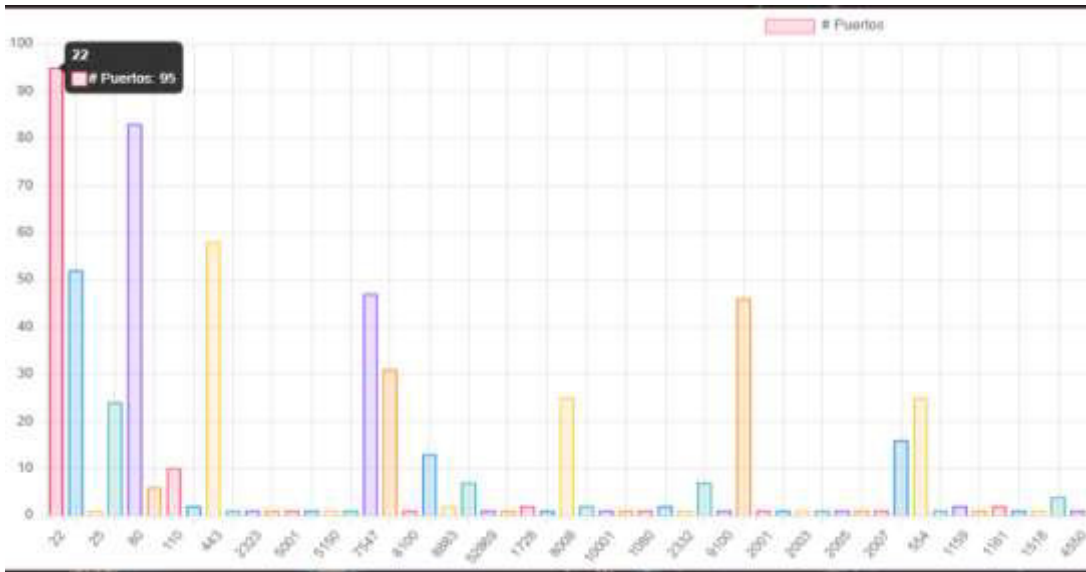


Fig. 38: Puerto 22 en Guayas

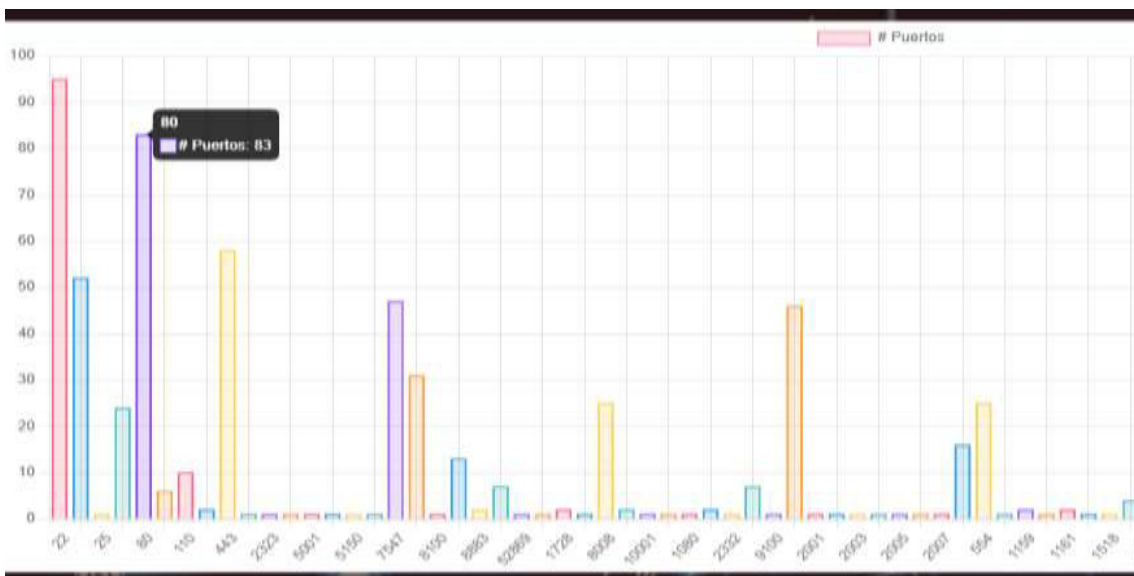


Fig. 39: Puerto 80 en Guayas

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Python es un lenguaje de programación de alto nivel orientado a objetos mediante el cual podemos desarrollar tanto sistemas Web como Aplicaciones informáticas, lo que lo hace factible es la gran variedad de usos que tiene gracias a sus librerías que permanentemente son actualizadas.
- MongoDB al ser una base de datos NoSQL es una herramienta que nos brinda rendimiento, flexibilidad y escalabilidad mismas que mantienen una consistencia y seguridad en los datos.
- La aplicación de la metodología Scrum permite al desarrollador tener una buena organización para cumplir con sus objetivos y satisfacer las necesidades del cliente.
- Flask es un microframework utilizado para el desarrollo del Sistema Web a partir de Python el cual nos permite un desarrollo ágil y rápido.
- Las pruebas funcionales permiten saber si el sistema o herramienta cumple con los requerimientos del cliente.

4.2 Recomendaciones

- Se debe tener en cuenta que al usar Python tenemos que estar pendientes a las actualizaciones de sus librerías y de Python como tal para no perder el funcionamiento tanto del Script como del Sistema Web.
- En el Sistema Web únicamente el o los administradores del sistema listo podrán activar las cuentas de los nuevos clientes registrados.
- Para el desarrollo de nuevas funcionalidades tanto en el Sistema Web como en el Script se debe tener en cuenta la compatibilidad de los sistemas, librerías necesarias a usar y un previo análisis de los nuevos requerimientos.

5. REFERENCIAS

- [1] I. Wignore, «techtarget,» 02 09 2017. [En línea]. Available: <https://searchdatacenter.techtarget.com/es/definicion/Internet-de-las-cosas-IoT>. [Último acceso: 19 11 2019].
- [2] INCIBE, «incibe-cert,» 25 04 2019. [En línea]. Available: <https://www.incibe-cert.es/blog/importancia-seguridad-iot-principales-amenazas>. [Último acceso: 06 11 2019].
- [3] K. L. Lueth, «IOT ANALYTICS,» 08 08 2018. [En línea]. Available: <https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b>. [Último acceso: 30 10 2019].
- [4] V. Chamorro y R. Rivera, «Twitter mining for multiclass classification events of traffic and pollution,» de *International Conference on Human Systems Engineering and Design: Future Trends and Applications*, Munich, 2019.
- [5] L. Pazmiño, F. Flores, L. Ponce, J. Zaldumbide, V. Parraga, B. Loarte, G. Cevallos, I. Maldonado and R. Rivera, "Challenges and Opportunities of IoT Deployment in Ecuador," in *2019 International Conference on Information Systems and Software Technologies (ICI2ST)*, Quito, 2019.
- [6] E. V. Cruz, 13 05 2016. [En línea]. Available: <https://www.edgarvasquez.com/Vulnerabilidad-internet-de-las-cosas>. [Último acceso: 12 11 2019].
- [7] L. P. Corrales, «BIBDIGITAL-EPN,» 05 10 2017. [En línea]. Available: <https://bibdigital.epn.edu.ec/handle/15000/10020>. [Último acceso: 20 11 2019].
- [8] R. R. Guevara, ANÁLISIS DE CARACTERÍSTICAS ESTÁTICAS DE FICHEROS EJECUTABLES PARA LA CLASIFICACIÓN DE MALWARE, UNIVERSIDAD POLITÉCNICA DE MADRID, 2014.

- [9] «Csirt-cv,» 23 12 2014. [En línea]. Available: <http://www.csirtcv.gva.es/es/noticias/csirt-cv-publica-el-informe-%E2%80%9Cseguridad-en-internet-de-las-cosas%E2%80%9D.html>. [Último acceso: 01 11 2019].
- [10] R. Rivera, L. Pazmiño, F. Becerra y J. Barriga, «An Analysis of Cyber Espionage Process,» de *Developments and Advances in Defense and Security. Proceedings of MICRADS 2021*, Cartagena, 2021.
- [11] J. Albors, «ESET,» 25 07 2018. [En línea]. Available: <https://www.welivesecurity.com/la-es/2018/07/25/seguridad-iot-a-tiempo-ganar-batalla>. [Último acceso: 22 10 2019].
- [12] C. Jiménez y R. Rivera, «Ciberseguridad del IoT: Un Análisis en Países de la Unión Europea,» *Revista Ibérica de Sistemas e Tecnologias de Informação*, nº E39, pp. 461-476, 2021.
- [13] R. R. Guevara, Tools for the detection and analysis of potentially unwanted programs, (Doctoral dissertation, Tesis doct. Nov. de 2018. doi: 10.20868/UPM.thesis.53395), 2018.
- [14] M. Ponce, «Ecuador Estado Digital Ene/20,» Mentinno - Innovation & Lifetime Value Partners, Quito, Ecuador, 2020.
- [15] S. L. Karen, «La internet de las cosas breve reseña,» internet society, Geneva, Switzerland, 2015.
- [16] J. F. Pareja Quinaluisa, «Evaluación de procesos de software utilizando EvalProSoft Aplicado a un caso de estudio,» 08 02 2012. [En línea]. Available: <http://bibdigital.epn.edu.ec/handle/15000/4491> .
- [17] P. Galiana, «IEBS,» 20 04 2021. [En línea]. Available: <https://www.iebschool.com/blog/metodologia-scrum-agile-scrum/>.
- [18] J. Roche, «Deloitte,» 2021. [En línea]. Available: <https://www2.deloitte.com/es/es/pages/technology/articles/roles-y-responsabilidades-scrum.html>.

- [19] «Softeng,» 2021. [En línea]. Available: <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html>.
- [20] «VIEWNEXT,» 2021. [En línea]. Available: <https://www.viewnext.com/artefactos-scrum/>. [Último acceso: 2021].
- [21] M. Rehkopf, «Atlassian,» 2021. [En línea]. Available: <https://www.atlassian.com/es/agile/project-management/user-stories>.
- [22] «Deloitte,» 2021. [En línea]. Available: <https://www2.deloitte.com/es/es/pages/technology/articles/artefactos-scrum.html>.
- [23] A. Pérez, «ObsBusiness,» 2021. [En línea]. Available: <https://www.obsbusiness.school/blog/las-5-etapas-en-los-sprints-de-un-desarrollo-scrum>.
- [24] E. R. Frías, «Unweaving the web,» [En línea]. Available: <https://estebanromero.com/herramientas-emprender-desarrollar-proyectos/marvel-una-herramienta-para-realizar-prototipos/>.
- [25] «Universidad de Alicante,» [En línea]. Available: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>. [Último acceso: 10 07 2021].
- [26] «epitech,» 08 07 2021. [En línea]. Available: <https://www.epitech-it.es/flask-python>.
- [27] R. Ángel, «OpenWebinars,» 23 Septiembre 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-python/>.
- [28] J. Ferrero, «Fixed Buffer,» 24 Marzo 2020. [En línea]. Available: <https://www.fixedbuffer.com/como-crear-una-base-de-datos-en-alta-disponibilidad-con-mongodb/>.
- [29] Isaac, «DesdeLinux,» 02 Marzo 2021. [En línea]. Available: <https://blog.desdelinux.net/github-vs-gitlab/>.

- [30] R. De la Vega, «Pharos,» 26 Enero 2021. [En línea]. Available: <https://pharos.sh/integrar-mongodb-con-python-usando-pymongo/>.
- [31] K. Tran, «Towards Data Science,» 13 Enero 2021. [En línea]. Available: <https://towardsdatascience.com/stop-using-print-to-debug-in-python-use-icecream-instead-79e17b963fcc>.
- [32] D. G. Ionos, «Digital Guide Ionos,» 07 Julio 2021. [En línea]. Available: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/logging-de-python/>.
- [33] R. Fernandez, «Unipython,» 17 Abril 2020. [En línea]. Available: <https://unipython.com/programacion-de-redes-en-python-sockets/>.
- [34] Dnspython, «Dnspython,» 05 Julio 2020. [En línea]. Available: <https://www.dnspython.org/about/>.
- [35] W. Mohammad, «Browserstack,» 11 Mayo 2021. [En línea]. Available: <https://www.browserstack.com/guide/take-screenshot-with-selenium-python>.
- [36] A. Calles, «FluProject,» 8 Diciembre 2016. [En línea]. Available: <https://www.flu-project.com/2016/12/construyendo-nuestro-propio-escaner-de.html>.
- [37] O. Espinosa, «RedesZone,» 9 Diciembre 2019. [En línea]. Available: <https://www.redeszone.net/tutoriales/internet/que-es-whois/>.
- [38] mongoDB, «mongoDB,» 2021. [En línea]. Available: <https://www.mongodb.com/es/what-is-mongodb>. [Último acceso: 27 08 2021].
- [39] «Globe,» 2018. [En línea]. Available: <https://www.globetesting.com/pruebas-funcionales>. [Último acceso: 27 08 2021].
- [40] «PMOinformatica,» 08 08 2016. [En línea]. Available: <http://www.pmoinformatica.com/2016/08/pruebas-aceptacion-software-istqb.html>. [Último acceso: 27 08 2021].
- [41] «myservername.com,» 2021. [En línea]. Available: <https://es.myservername.com/what-is-software-compatibility-testing#:~:text=La%20prueba%20de%20compatibilidad%20consiste,manera%2>

0para%20todas%20las%20plataformas.&text=Por%20lo%20general%2C%20el
%20equipo,aplicaci%C3%B3n%20en%20una%20%C3%BAnica%20plataform.

- [42] I. Learning, «InGenio Learning,» 25 Enero 2021. [En línea]. Available: <https://ingenio.edu.pe/que-es-django-y-como-puede-facilitar-tus-labores/>.
- [43] «Capacítate,» [En línea]. Available: <https://www.capacitarte.org/blog/nota/que-es-y-para-que-sirve-python>. [Último acceso: 21 09 2021].
- [44] J. D. Muñoz, «OpenWebinars,» 17 11 2017. [En línea]. Available: <https://openwebinars.net/blog/que-es-flask/>. [Último acceso: 21 09 2021].

6. ANEXOS

6.1 Manual Técnico

- Recopilación de Requerimientos
- Historias de Usuario
- Sprint Backlog
- Diseño de Interfaces
- Pruebas de compatibilidad
- Pruebas de aceptación

6.2 Manual de Usuario

<https://www.youtube.com/watch?v=z29qh0N2PYI>

6.3 Manual de Instalación

https://github.com/EdisonStalin/Flask_lotEcuador

MANUAL TÉCNICO

**DESARROLLO DE UNA HERRAMIENTA DE SOFTWARE PARA
EL ANÁLISIS DE VULNERABILIDADES EN DISPOSITIVOS IOT
EN ECUADOR.**

EDISON STALIN JUMBO JUMBO

edison.jumbo@epn.edu.ec

JEFFERSON RICARDO LLUMIQUINGA LLUMIQUINGA

jefferson.llumiquinga@epn.edu.ec

ÍNDICE DE CONTENIDOS

1. Levantamiento de requerimientos.....	3
2. Historias de Usuario	4
3. Sprint Backlog	10
4. Diseño de interfaces	16
4.1 Sistema Web	16
5. Diseño de base de datos	19
6. Pruebas de compatibilidad.....	22
6.1 Sistema Web	22
7. pruebas de aceptación.....	23
7.1 Sistema Web	23

1. LEVANTAMIENTO DE REQUERIMIENTOS

Mediante los estudios realizados podemos definir una lista de requerimientos iniciales los cuales nos sirve para el desarrollo de la herramienta, como presentamos en la **TABLA I.**

TABLA I: Levantamiento de requerimientos

Identificación de Requerimiento	Descripción.
RR001	Como usuario administrador necesito realizar una búsqueda de direcciones IPv4.
RR002	Como usuario administrador necesito guardar las direcciones IPv4 en MongoDB.
RR003	Como usuario administrador necesito actualizar los datos.
RR004	Como usuario administrador necesito guardar capturas de pantalla de puertos abiertos.
RR005	Como usuario final necesito visualizar una página informativa en el sistema web.
RR006	Como usuario final se necesita registrarse en el sistema.
RR007	Como usuario administrador necesito iniciar y cerrar sesión.
RR008	Como usuario administrador necesito realizar la activación de cuentas de los usuarios registrados por primera vez.
RR009	Como usuario final necesito realizar lo siguiente en el sistema web. Visualizar las direcciones IPv4 encontradas
RR010	Como usuario final necesito realizar lo siguiente en el sistema web. Visualizar a detalle los datos de cada dirección IPv4.
RR011	Como usuario final necesito realizar lo siguiente en el sistema web. Búsqueda y filtrado de direcciones IPv4. <ul style="list-style-type: none">• Dirección IPv4.• Puerto.

	<ul style="list-style-type: none"> • Ciudad.
RR012	Como usuario final necesito visualizar los resultados de la búsqueda.
RR013	Como usuario final necesito visualizar un análisis de puertos por ciudad.

2. HISTORIAS DE USUARIO

Al finalizar la etapa del levantamiento de requerimientos y el análisis respectivo se realizaron las historias de usuario las cuales se muestran desde la **TABLA II** hasta la **TABLA XV**.

TABLA II: Historia de Usuario 2 - Buscar direcciones IPv4

Historia de Usuario	
Identificador (ID): HU002	Usuario: Administrador
Nombre Historia: Buscar direcciones IPv4	
Prioridad en negocio: Alta	Riesgo de desarrollo: Medio
Iteración Asignada: 1	
Responsable(es): Edison Jumbo-Jefferson Llumiquinga	
Descripción: El proceso de búsqueda para direcciones IPv4 se da de manera aleatoria y dentro del rango de direcciones en territorio ecuatoriano.	
Observación: Únicamente el administrador del sistema puede correr el script y buscar tantas direcciones IPv4 como desee.	

TABLA III: Historia de Usuario 3 - Recolectar direcciones IPv4

Historia de Usuario	
Identificador (ID): HU003	Usuario: Administrador
Nombre Historia: Recolectar direcciones IPv4	
Prioridad en negocio: Alta	Riesgo de desarrollo: Medio
Iteración Asignada: 2	
Responsable(es): Edison Jumbo-Jefferson Llumiquinga	
Descripción: La recolección de direcciones IPv4, con su respectiva información se da mediante MongoDB.	

Observación:

En MongoDB se guardan todas las direcciones IPv4 con o sin puertos abiertos, que se encuentren en el rango que pertenece a Ecuador.

TABLA IV: Historia de Usuario 4 - Actualizar direcciones IPv4

Historia de Usuario	
Identificador (ID): HU004	Usuario: Administrador
Nombre Historia: Actualizar direcciones IPv4	
Prioridad en negocio: Alta	Riesgo de desarrollo: Medio
Iteración Asignada: 1	
Responsable(es): Edison Jumbo-Jefferson Llumiquinga	
Descripción: La actualización de las direcciones IPv4 se da a partir del script y en MongoDB.	
Observación: Pasado los 30 días de inactividad por parte de la dirección IPv4 con estado (True) y los 15 días de inactividad por parte de la dirección IPv4 con estado (False) se procede hacer una nueva búsqueda de puertos.	

TABLA V: Historia de Usuario 5 - Capturar pantalla de puertos abiertos

Historia de Usuario	
Identificador (ID): HU005	Usuario: Administrador
Nombre Historia: Capturar pantalla de puertos abiertos	
Prioridad en negocio: Alta	Riesgo de desarrollo: Medio
Iteración Asignada: 1	
Responsable(es): Edison Jumbo-Jefferson Llumiquinga	
Descripción: Capturas de pantalla a las páginas web de puertos especificados en el Script.	
Observación: Mediante el navegador Google Chrome se abrirá la dirección IPv4 encontrada, el puerto específico y se toma una captura de pantalla de la página web encontrada.	

TABLA VI: Historia de Usuario 6 - Visualizar página informativa en el Sistema Web

Historia de Usuario	
Identificador (ID): HU006	Usuario: Cliente
Nombre Historia: Visualizar página informativa en el Sistema Web	

Prioridad en negocio: Alta	Riesgo de desarrollo: Medio
Iteración Asignada: 4	
Responsable(es): Edison Jumbo-Jefferson Llumiquinga	
Descripción: Página de inicio de sesión del sistema web.	
Observación: Los usuarios pueden visualizar la página de inicio donde encontramos información acerca de direcciones IPv4 activas por provincia, aparatos que se pueden conectar a Wifi e información acerca de nosotros.	

TABLA VII: Historia de Usuario 7 – Registrar usuarios

Historia de Usuario	
Identificador (ID): HU007	Usuario: Cliente
Nombre Historia: Registrar usuarios	
Prioridad en negocio: Alta	Riesgo de desarrollo: Medio
Iteración Asignada: 4	
Responsable(es): Edison Jumbo-Jefferson Llumiquinga	
Descripción: Necesita registrarse para acceder al dashboard y sus funcionalidades.	
Observación: Al momento de registrarse no puede iniciar sesión inmediatamente ya que necesita la activación de su cuenta por parte de un administrador.	

TABLA VIII: Historia de Usuario 8 - Editar Cuenta

Historia de Usuario	
Identificador (ID): HU008	Usuario: Cliente
Nombre Historia: Editar cuenta	
Prioridad en negocio: Media	Riesgo de desarrollo: Medio
Iteración Asignada: 4	
Responsable(es): Edison Jumbo-Jefferson Llumiquinga	
Descripción: El usuario podrá editar su información personal y su contraseña.	
Observación: Esta actualización de datos se dará únicamente cuando haya iniciado sesión y comprobada la identidad.	

TABLA IX: Historia de Usuario 9 - Iniciar y cerrar sesión de Administrador

Historia de Usuario	
Identificador (ID): HU009	Usuario: Administrador
Nombre Historia: Iniciar y cerrar sesión de Administrador	
Prioridad en negocio: Alta	Riesgo de desarrollo: Medio
Iteración Asignada: 3	
Responsable(es): Edison Jumbo-Jefferson Llumiquinga	
Descripción: El administrador tiene que iniciar sesión para poder ver los nuevos usuarios registrados y verificar su información previo a la activación de la cuenta.	
Observación: El rol de administrador es asignado por el equipo de desarrollo.	

TABLA X: Historia de Usuario 10 – Activar cuenta

Historia de Usuario	
Identificador (ID): HU010	Usuario: Administrador
Nombre Historia: Activar cuenta	
Prioridad en negocio: Alta	Riesgo de desarrollo: Medio
Iteración Asignada: 3	
Responsable(es): Edison Jumbo-Jefferson Llumiquinga	
Descripción: Permiso por parte del administrador para que el usuario pueda iniciar sesión.	
Observación: Un usuario no tiene permiso para iniciar sesión mientras el administrador no haya activado su cuenta.	

TABLA XI: Historia de Usuario 11 - Visualizar direcciones IPv4 encontradas

Historia de Usuario	
Identificador (ID): HU011	Usuario: Cliente
Nombre Historia: Visualizar direcciones IPv4 encontradas	
Prioridad en negocio: Alta	Riesgo de desarrollo: Medio
Iteración Asignada: 4	
Responsable(es): Edison Jumbo-Jefferson Llumiquinga	
Descripción:	

El cliente puede encontrar en una tabla todas las direcciones IPv4 recolectadas en MongoDB.

Observación:

En la tabla únicamente se encuentra las direcciones IPv4 con puertos abiertos.

TABLA XII: Historia de Usuario 12 - Visualizar los detalles de cada dirección IPv4

Historia de Usuario	
Identificador (ID): HU012	Usuario: Cliente
Nombre Historia: Visualizar los detalles de cada dirección IPv4	
Prioridad en negocio: Alta	Riesgo de desarrollo: Medio
Iteración Asignada: 4	
Responsable(es): Edison Jumbo-Jefferson Llumiquinga	
Descripción: Cada dirección IPv4 tiene un enlace para observar el resto de información con la que cuenta dicha dirección.	
Observación: La información más relevante en los detalles de la Dirección IPv4 son las capturas de pantalla y el banner.	

TABLA XIII: Historia de Usuario 13 – Buscar y filtrar direcciones IPv4

Historia de Usuario	
Identificador (ID): HU013	Usuario: Cliente
Nombre Historia: Buscar y filtrar direcciones IPv4	
Prioridad en negocio: Alta	Riesgo de desarrollo: Medio
Iteración Asignada: 4	
Responsable(es): Edison Jumbo-Jefferson Llumiquinga	
Descripción: Proceso de búsqueda de direcciones IPv4 en el Sistema Web.	
Observación: El proceso de búsqueda consta de 3 tipos de filtrado: Dirección IPv4, Puerto y Ciudad.	

TABLA XIV: Historia de Usuario 14 - Visualizar los resultados de la búsqueda

Historia de Usuario	
Identificador (ID): HU014	Usuario: Cliente
Nombre Historia: Visualizar los resultados de la búsqueda	

Prioridad en negocio: Alta	Riesgo de desarrollo: Medio
Iteración Asignada: 4	
Responsable(es): Edison Jumbo-Jefferson Llumiquinga	
Descripción: Respuesta a la petición del usuario.	
Observación: Se mostrará la información en la tabla según la opción de búsqueda que el usuario haya realizado.	

TABLA XV: Analizar los puertos por ciudad

Historia de Usuario	
Identificador (ID): HU015	Usuario: Cliente
Nombre Historia: Analizar los puertos por ciudad	
Prioridad en negocio: Alta	Riesgo de desarrollo: Medio
Iteración Asignada: 4	
Responsable(es): Edison Jumbo-Jefferson Llumiquinga	
Descripción: Al momento de hacer una búsqueda por ciudad mostramos una gráfica donde observamos el número de puertos existentes en cada ciudad.	
Observación: El análisis únicamente se da en relación a la ciudad con los puertos activos.	

3. SPRINT BACKLOG

A continuación, se presenta los detalles de los *Sprint* de desarrollo que han sido implementados tanto en el *Script* como en el Sistema Web, como se muestra en la **Tabla XVI**

Tabla XVI. Elaboración de sprint Backlog

Elaboración del Sprint Backlog						
ID-SB	NOMBRE	MÓDULO	ID-HU	HISTORIAS DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB000	Configuración del ambiente de desarrollo	Configuración del ambiente de desarrollo	N/A	N/A	<ul style="list-style-type: none"> Definir los requerimientos tanto para el desarrollo del Script como para el Sistema Web. Estudio del uso de la base de datos NoSQL MongoDB. Configuración de las herramientas de desarrollo. Configuración del ambiente de desarrollo tanto para el Script como para el Sistema Web. 	20h
SB001	Búsqueda de direcciones IPv4	Búsqueda	HU002	Buscar direcciones Ipv4	<ul style="list-style-type: none"> Investigar el rango de direcciones IPv4 que son válidos en Ecuador. Buscar y descargar archivo dat que contenga direcciones IP con su respectiva ubicación. 	

					<ul style="list-style-type: none"> • Creamos la función para encontrar direcciones Ipv4 de Ecuador aleatoriamente. • Comprobar que la dirección Ipv4 encontrada es de Ecuador. • Definimos una condicional donde nos valide que el código del país en el caso del Sistema tiene que ser "EC". 	120h
			HU004	Actualizar direcciones Ipv4	<ul style="list-style-type: none"> • Utilizamos la misma función de búsqueda para verificar la fecha de registro de la dirección Ipv4. • Verificar si la dirección Ipv4 ha tenido una inactividad entre 15 a 30 días establecidos, se procede a la actualización de los datos de la dirección Ipv4. 	
			HU005	Capturar pantalla de puertos abiertos	<ul style="list-style-type: none"> • Encontrar un archivo ejecutable que nos ayuda abrir la dirección Ipv4 junto con el puerto abierto en el navegador Google Chrome para poder tomar la captura de pantalla. • Crear una función donde ocupemos el archivo ejecutable y poder guardar el nombre de la imagen en MongoDB. 	

SB002	Recolección de datos	Guardar datos en MongoDB	HU003	Recolectar direcciones Ipv4	<ul style="list-style-type: none"> • Crear una cuenta en MongoDB única para este proyecto. • Especificar los parámetros de desarrollo acorde a la necesidad del sistema. • Configuración de los clientes que tengan acceso a la base de datos. • Instalación librería pymongo para la conexión entre MongoDB y el Script. • Creamos una función donde especificamos el MongoClient que es la línea de conexión entre MongoDB y el Script. • Creamos una función para buscar direcciones Ipv4 en MongoDB, si ya existe la dirección Ipv4 el Script debe continuar la búsqueda hasta cumplir cantidad de direcciones especificadas al inicio. • Creamos una función para poder insertar las direcciones Ipv4 encontradas. • Se generan filtros y condiciones para verificar la dirección Ipv4 y poder añadir todos los datos a MongoDB. 	120h
-------	----------------------	--------------------------	-------	-----------------------------	---	------

SB003	Inicio de sesión usuario Administrador		HU009	Iniciar y cerrar sesión de Administrador	<ul style="list-style-type: none"> • Diseñar interfaz de registro e inicio de sesión. • Dar el rol administrador por parte de los desarrolladores. • Crear las funciones para autenticación de usuarios. • Validación de campos y roles. • Crear la función para cerrar sesión en el Sistema Web. 	80h
			HU010	Activar cuenta	<ul style="list-style-type: none"> • Diseño de la interfaz para activación de nuevos usuarios registrados. • Crear la función para la aprobación de las cuentas. • Crear la función de correo para notificar la activación de la cuenta. 	
			HU001	Iniciar sesión de usuario	<ul style="list-style-type: none"> • Definir los campos necesarios para el inicio de sesión. • Diseño de interfaz para el inicio de sesión en el Sistema Web. • Crear funciones para autenticación de usuarios • Validación de campos. 	120

SB004	Inicio de sesión del usuario cliente		HU006	Visualizar página informativa en el Sistema Web	<ul style="list-style-type: none"> • Diseño de interfaz de la página de inicio del Sistema Web. • Visualizar información acerca de IoT.
			HU007	Registrar usuarios	<ul style="list-style-type: none"> • Diseñar la interfaz para el registro de usuarios. • Crear funciones para registro de usuarios. • Definir los campos necesarios para el registro de usuarios. • Validación de campos.
			HU008	Editar perfil	<ul style="list-style-type: none"> • Diseñar la interfaz para edición los datos del usuario y contraseña. • Crear funciones para la modificación de los usuarios. • Validación de campos.
			HU011	Visualizar direcciones Ipv4 encontradas	<ul style="list-style-type: none"> • Diseñar la interfaz para poder observar las direcciones Ipv4. • Crear funciones para poder visualizar las direcciones Ipv4 almacenadas en MongoDB.
			HU012	Visualizar los detalles de	<ul style="list-style-type: none"> • Diseñar la interfaz para poder observar los detalles y datos de cada dirección Ipv4.

				<p>cada dirección Ipv4</p> <ul style="list-style-type: none"> • Crear funciones que nos permita observar los detalles. 	
			HU013	<p>Buscar y filtrar direcciones Ipv4</p> <ul style="list-style-type: none"> • Diseñar la interfaz para poder hacer la búsqueda de direcciones Ipv4. • Crear funciones para poder filtrar las búsquedas. 	
			HU014	<p>Visualizar los resultados de la búsqueda</p> <ul style="list-style-type: none"> • Diseñar la interfaz para poder visualizar las direcciones encontradas en la búsqueda. • Crear funciones para poder visualizar los resultados de las búsquedas. 	
			HU015	<p>Analizar puertos por ciudad</p> <ul style="list-style-type: none"> • Diseñar la interfaz para poder observar un análisis de puertos por ciudad. 	
SB005	Pruebas de Software			<p>Pruebas del Sistema Web y Script</p> <ul style="list-style-type: none"> • Pruebas de compatibilidad • Pruebas de aceptación 	20h
TOTAL 480 HORAS					

4. DISEÑO DE INTERFACES

A continuación, se presentan los prototipos que se han realizado en Marvel previo al desarrollo del Sistema Web, donde se muestra las características que tiene cada una de las interfaces.

4.1 Sistema Web

En la **Fig. 1** se muestra un boceto del como se muestra la página de inicio del Sistema Web



Fig. 1: Página de inicio

En la **Fig. 2** se muestra el boceto del como se quiere que se muestre el formulario de inicio de sesión tanto para el usuario final como para el administrador.



Fig. 2: Inicio de sesión

En la **Fig. 3** se muestra el boceto del formulario para el registro de usuarios

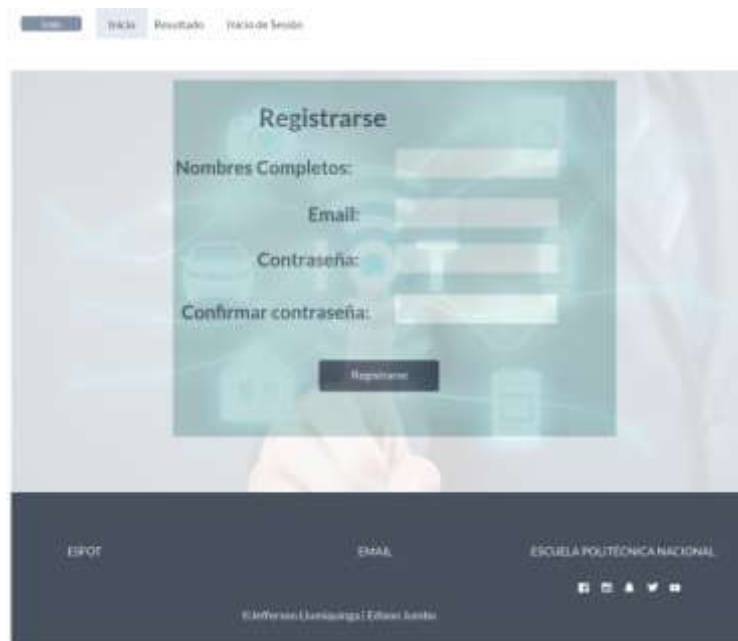


Fig. 3: Registro de usuario

En la **Fig. 4** se muestra el boceto del como se desea que se muestre la información acerca de las direcciones IPv4 encontradas con sus respectivos puertos y el análisis por provincias.



Fig. 4: Visualizar direcciones IPv4 y análisis de puertos

En la **Fig. 5** se muestra el boceto para la opción de edición de información del usuario registrado.

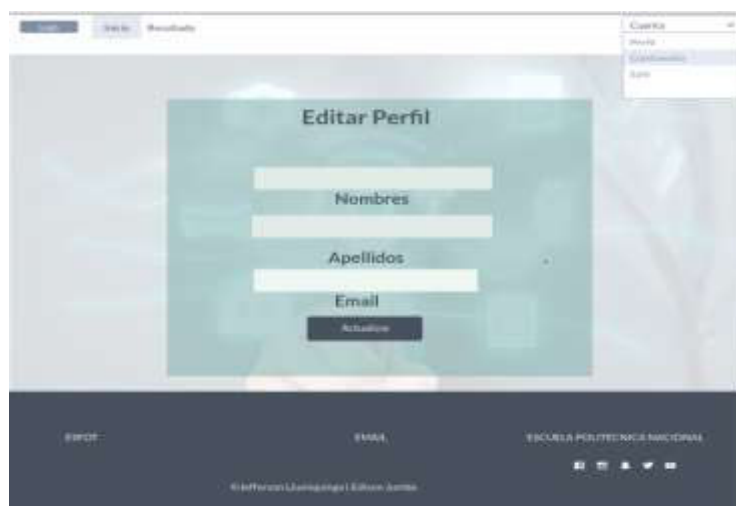


Fig. 5: Editar perfil de usuario

En la **Fig. 6** se muestra el boceto de la opción para que el usuario pueda editar su contraseña si lo ve necesario.



Fig. 6: Editar contraseña de usuario

En la **Fig. 7** el panel donde el administrador va activar las cuentas de los nuevos usuarios registrados.

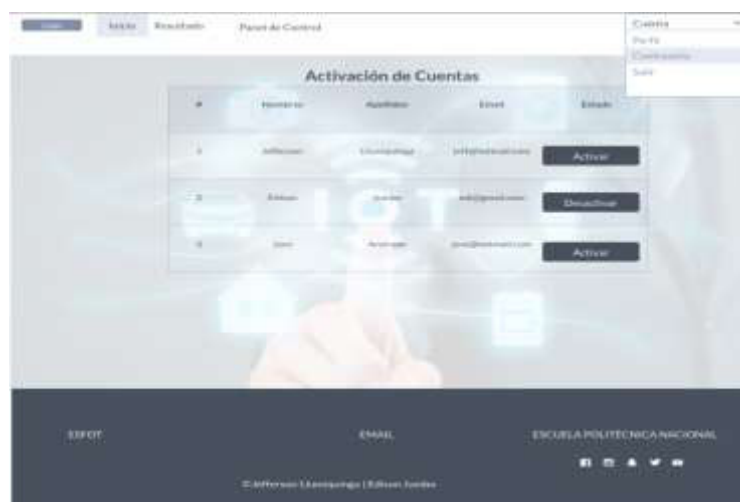


Fig. 7: Activación de cuenta por parte de Administrador

5. DISEÑO DE BASE DE DATOS

MongoDB es una base de datos NoSQL que usa documentos para guardar la información requerida, como podemos ver en la **Fig. 8**, **Fig. 9** y **Fig. 10** nos muestra el documento diseñado para almacenar las direcciones IPv4 y su información mientras que en la **Fig. 11** y **Fig. 12** muestra el documento diseñado para el registro de usuarios.

iotecuador.Devices

COLLECTION SIZE: 10.31MB TOTAL DOCUMENTS: 24565 INDEXES TOTAL SIZE: 1.21MB

Find Indexes Schema Anti-Patterns 0 Aggregation Search Indexes ●

Fig. 8: Nombre del documento y cantidad de datos almacenados

```
_id: ObjectId("60edc4820013b80f4826e0f7")
Direccion: "191.100.20.14"
Estado: true
Fecha: "2021-08-27 16:28:05"
Whois: Object
  nir: null
  asn_registry: "lacnic"
  asn: "27668"
  asn_cidr: "191.100.16.0/21"
  asn_country_code: "EC"
  asn_date: "2014-02-03"
  asn_description: "ETAPA EP, EC"
  query: "191.100.20.14"
  nets: Array
    raw: null
    referral: null
    raw_referral: null
  Dominio: "14.191-100-20.etapanet.net"
  Dns: "14.20.100.191.in-addr.arpa."
```

Fig. 9: Documento devices

```

  puerto: Array
    > 0: Object
    > 1: Object
      Puerto: "2000"
      Banner: "\x01\x00\x00\x"
      Imagen: "Noimagen.png"
    > 2: Object
      Puerto: "888"
      Banner: "None"
      Imagen: "Noimagen.png"
    > 3: Object
      Puerto: "8291"
      Banner: "None"
      Imagen: "Noimagen.png"
  Localizacion: Object
    dma_code: 0
    area_code: 0
    metro_code: null
    postal_code: null
    country_code: "EC"
    country_code3: "ECU"
    country_name: "Ecuador"
    continent: "SA"
    region_code: "02"
    city: "Cuenca"
    latitude: -2.8832999999999913
    longitude: -78.9833
    time_zone: "America/Guayaquil"

```

Fig. 10: Documento devices

iotecuador.User

COLLECTION SIZE: 1.38KB TOTAL DOCUMENTS: 5 INDEXES TOTAL SIZE: 56KB

Find Indexes Schema Anti-Patterns 0 Aggregation Search Indexes ●

Fig. 11: Nombre del documento y cantidad de datos almacenados

```

_id: ObjectId("612e7a4eb293d0977323bc04")
last_login: 2021-09-27T12:37:32.732+00:00
username: null
first_name: "Edison"
last_name: "Jumbo"
msg: "Proyecto de la U"
email: "edisonsjumbo@gmail.com"
password: Binary('JDJiJDEyJHBMbTFuOW5ibFAUZDRhLmhqWEJUTC5DSklPa0lteERZczdrRXZ3TGs1RFIvcUl0UwtSOW51', 0)
is_active: true
date_joined: 2021-08-31T13:51:58.742+00:00
role: "user"

```



Fig. 12: Documento user

6. PRUEBAS DE COMPATIBILIDAD

6.1 Sistema Web

A continuación, desde la **¡Error! No se encuentra el origen de la referencia.** a la **¡Error! No se encuentra el origen de la referencia.**, se muestran las imágenes de prueba de compatibilidad realizado al Sistema Web en los navegadores, Google Chrome, Mozilla Firefox y Microsoft Edge.

Google Chrome



Fig. 13: Prueba de compatibilidad – Navegador Google Chrome



Fig. 14: Prueba de compatibilidad – Navegador Mozilla Firefox

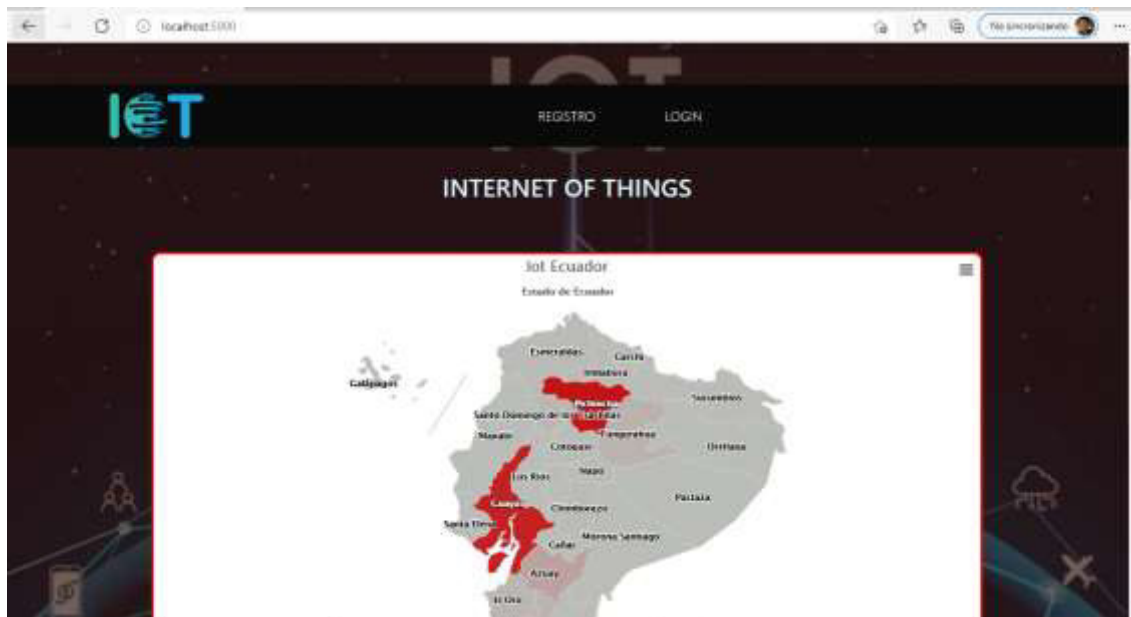


Fig. 15: Prueba de compatibilidad – Navegador Microsoft Edge

7. PRUEBAS DE ACEPTACIÓN

Las pruebas de aceptación son realizadas para medir el grado de confianza en un sistema por lo tanto a continuación se muestra una serie de pruebas para verificar el correcto funcionamiento del Sistema Web.

7.1 Sistema Web

TABLA XVII: Prueba de aceptación - Búsqueda de direcciones IPv4

Prueba de Aceptación
Identificador (ID): PA002
Identificador historia de Usuario: HU002
Nombre para prueba de aceptación: Búsqueda de direcciones IPv4.
Descripción: Búsqueda de direcciones IPv4 en Ecuador a partir del Script.
Pasos de ejecución: Correr el Script. En el menú elegir la opción Analizar direcciones IPv4 en Ecuador. Escribir la cantidad de direcciones que desea analizar.
Resultado deseado: El Script comienza el análisis para encontrar direcciones IPv4 que se encuentren en el rango especificado en territorio ecuatoriano.
Evaluación de la prueba: Se comprueba el resultado esperado. Aprobación del cliente 100%.

TABLA XVIII: Prueba de aceptación - Recolección de direcciones IPv4

Prueba de Aceptación
Identificador (ID): PA003
Identificador historia de Usuario: HU00
Nombre para prueba de aceptación: Recolección de direcciones IPv4.
Descripción: Guardar direcciones IPv4 encontradas en MongoDB
Pasos de ejecución: Análisis de direcciones IPv4. Búsqueda de puertos abiertos en cada dirección IPv4. Guardar en MongoDB.
Resultado deseado: En MongoDB se guardan todas las direcciones IPv4 con o sin puertos abiertos, que se encuentren en el rango que pertenece a Ecuador.
Evaluación de la prueba: Se comprueba el resultado esperado. Aprobación del cliente 100%.

TABLA XIX: Prueba de aceptación - Actualización de direcciones IPv4

Prueba de Aceptación
Identificador (ID): PA004
Identificador historia de Usuario: HU004
Nombre para prueba de aceptación: Actualización de direcciones IPv4.
Descripción: Si se encuentra inactividad mayor a 30 días de la dirección IPv4 con estado (True) y 15 días de la dirección IPv4 con estado (False) guardada en MongoDB, el Script analiza dicha dirección y si cumple con la condición dada actualiza la dirección IPv4 y los nuevos puertos.
Pasos de ejecución: Al correr el Script se analiza el tiempo de inactividad de una dirección IPv4. Si ha sobrepasado los 30 días o 15 días de inactividad se actualizara la nueva dirección IPv4 con su correspondiente información.
Resultado deseado: En MongoDB no se pueden guardar dos veces una misma dirección IPv4, es por eso que a parte de la condición de inactividad también se cuenta con la condición que la dirección IPv4 debe ser única.
Evaluación de la prueba: Se comprueba el resultado esperado. Aprobación del cliente 100%.

TABLA XX: Prueba de aceptación - Captura de pantalla de puertos abiertos

Prueba de Aceptación
Identificador (ID): PA005

Identificador historia de Usuario: HU005
Nombre para prueba de aceptación: Captura de pantalla de puertos abiertos.
Descripción: Un ejecutable de Google Chrome nos da la opción de hacer capturas de pantalla según las condiciones dadas.
Pasos de ejecución: Análisis de puertos en cada dirección IPv4. La dirección IPv4 tendrá X cantidades de puertos y de cada uno se tomará una captura de pantalla. Las capturas tendrán el nombre de la dirección IPv4 y el puerto específico.
Resultado deseado: Las capturas se guardan en el directorio captura que se encuentra en el proyecto del Sistema Web.
Evaluación de la prueba: Se comprueba el resultado esperado. Aprobación del cliente 100%.

TABLA XXI: Prueba de Aceptación - Visualizar página informativa en el Sistema Web

Prueba de Aceptación
Identificador (ID): PA006
Identificador historia de Usuario: HU006
Nombre para prueba de aceptación: Visualizar página informativa en el sistema Web.
Descripción: Página de inicio informativa.
Pasos de ejecución: Visualizar el mapa donde nos muestra el número direcciones IPv4 con puertos activos que tiene cada provincia. Aparatos IoT. Acerca de nosotros. Aplicaciones IoT.
Resultado deseado: La página nos muestra información que pueden ver los usuarios sin la necesidad de iniciar sesión.
Evaluación de la prueba: Se comprueba el resultado esperado. Aprobación del cliente 100%.

TABLA XXII: Prueba de aceptación - Registro de usuarios

Prueba de Aceptación
Identificador (ID): PA007
Identificador historia de Usuario: HU007
Nombre para prueba de aceptación: Registro de usuarios.
Descripción: Registro de nuevos usuarios para uso del Sistema Web.
Pasos de ejecución: Ir a la dirección de registro de usuario desde el menú de navegación. Escribir los datos que se le solicita de manera correcta.

<p>Registrarse. Esperar la activación de su cuenta.</p>
<p>Resultado deseado: Si el usuario escribe todos sus datos de manera correcta no recibirá ningún mensaje de error y podrá esperar a que se le active su cuenta para que pueda iniciar sesión.</p>
<p>Evaluación de la prueba: Se comprueba el resultado esperado. Aprobación del cliente 100%.</p>

TABLA XXIII: Prueba de aceptación - Editar cuenta

Prueba de Aceptación
Identificador (ID): PA008
Identificador historia de Usuario: HU008
Nombre para prueba de aceptación: Editar Cuenta.
Descripción: El usuario tendrá la posibilidad de editar sus datos personales y su contraseña.
<p>Pasos de ejecución: Al ingresar al sistema tenemos en el menú de navegación una pestaña llamada cuenta. Podrá editar su información personal. Podrá editar su contraseña, primero se debe escribir la contraseña actual.</p>
<p>Resultado deseado: Tanto los datos personales como la contraseña se las puede editar de manera correcta siguiendo las validaciones previas como lo son que si desea cambiar el email debe llevar el @ y un punto y en las contraseñas llevar al menos una letra mayúscula, un carácter especial y un número.</p>
<p>Evaluación de la prueba: Se comprueba el resultado esperado. Aprobación del cliente 100%.</p>

TABLA XXIV: Prueba de aceptación - Iniciar y cerrar sesión usuario Administrador

Prueba de Aceptación
Identificador (ID): PA009
Identificador historia de Usuario: HU009
Nombre para prueba de aceptación: Iniciar y cerrar sesión usuario Administrador.
Descripción: El usuario administrador podrá iniciar y cerrar sesión de la misma manera como lo hace el usuario final.
<p>Pasos de ejecución: Dirigirse a la dirección login desde el menú de navegación. Escribir correo y contraseña para validar datos. Validación de campos. Ingreso al sistema.</p>
<p>Resultado deseado: Si los datos son correctos no se tendrá problema para entrar al sistema y podrá activar las cuentas de usuarios nuevos y también utilizar el dashboard del Sistema Web.</p>

<p>Evaluación de la prueba: Se comprueba el resultado esperado. Aprobación del cliente 100%.</p>

TABLA XXV: Prueba de aceptación - Activación de cuenta

Prueba de Aceptación
Identificador (ID): PA010
Identificador historia de Usuario: HU010
Nombre para prueba de aceptación: Activación de cuenta.
Descripción: El usuario administrador debe activar las cuentas de nuevos usuarios después hacer una pequeña revisión del porque desea usar el sistema.
<p>Pasos de ejecución: El usuario administrador inicia sesión. Se dirige al panel de control mediante el menú de navegación. Activa o desactiva las cuentas de los usuarios finales.</p>
<p>Resultado deseado: El administrador puede ingresar correctamente al Sistema Web para observar los usuarios que necesitan revisión.</p>
<p>Evaluación de la prueba: Se comprueba el resultado esperado. Aprobación del cliente 100%.</p>

TABLA XXVI: Prueba de aceptación - Visualizar direcciones IPv4 encontradas

Prueba de Aceptación
Identificador (ID): PA011
Identificador historia de Usuario: HU011
Nombre para prueba de aceptación: Visualizar direcciones IPv4 encontradas.
Descripción: En una tabla se muestran las direcciones IPv4, ciudad, fecha y puerto.
<p>Pasos de ejecución: Al iniciar sesión aparece en la barra de navegación el dashboard. Observamos información varia antes de la tabla. En la tabla se muestra únicamente las direcciones IPv4 que cuentan con puertos abiertos.</p>
<p>Resultado deseado: Para poder hacer uso del dashboard es necesario siempre haber iniciado sesión.</p>
<p>Evaluación de la prueba: Se comprueba el resultado esperado. Aprobación del cliente 100%.</p>

TABLA XXVII: Prueba de aceptación - Visualizar los detalles de cada dirección IPv4

Prueba de Aceptación
Identificador (ID): PA012
Identificador historia de Usuario: HU012
Nombre para prueba de aceptación: Visualizar los detalles de cada dirección IPv4.
Descripción: El usuario podrá ver la información extra de cada dirección IPv4.
Pasos de ejecución: Visualizar todas las direcciones IPv4 en la tabla que se encuentra en el dashboard. Click en cualquier dirección IPv4 que desee. Redirige a detalles de direcciones IPv4 donde se visualiza los datos.
Resultado deseado: Como una de las informaciones extra e importantes es el Banner donde nos puede mostrar el sistema operativo del equipo al cual pertenece la dirección IPv4.
Evaluación de la prueba: Se comprueba el resultado esperado. Aprobación del cliente 100%.

TABLA XXVIII: Prueba de aceptación - Búsqueda y filtrado de direcciones IPv4

Prueba de Aceptación
Identificador (ID): PA013
Identificador historia de Usuario: HU013
Nombre para prueba de aceptación: Búsqueda y filtrado de direcciones IPv4.
Descripción: Podemos hacer la búsqueda de cualquier dirección IPv4 y puerto que haya almacenado el sistema en MongoDB.
Pasos de ejecución: Escoger una de las opciones de búsqueda. Escribir la dirección IPv4, el puerto o ciudad según la opción que haya escogido. Visualizar la búsqueda.
Resultado deseado: Mediante los filtros de búsqueda se pudo mejorar la observación de direcciones IPv4.
Evaluación de la prueba: Se comprueba el resultado esperado. Aprobación del cliente 100%.

TABLA XXIX: Prueba de aceptación - Visualizar los resultados de la búsqueda

Prueba de Aceptación
Identificador (ID): PA014
Identificador historia de Usuario: HU014
Nombre para prueba de aceptación: Visualizar los resultados de la búsqueda.
Descripción: En la tabla que se encuentra en el dashboard se mostrara los resultados de la búsqueda.
Pasos de ejecución: Realizar la búsqueda deseada según las opciones indicadas anteriormente. Visualizar los resultados de la búsqueda dentro de la tabla.

<p>Resultado deseado: Las búsquedas se hacen con la finalidad de observar únicamente ciertas direcciones IPv4 o puertos en específico.</p>
<p>Evaluación de la prueba: Se comprueba el resultado esperado. Aprobación del cliente 100%.</p>

TABLA XXX: Prueba de aceptación - Análisis de puertos por ciudad

Prueba de Aceptación
Identificador (ID): PA015
Identificador historia de Usuario: HU015
Nombre para prueba de aceptación: Análisis de puertos por ciudad.
Descripción: Se observan datos estadísticos al momento de realizar la búsqueda por ciudad.
<p>Pasos de ejecución: Se realiza la búsqueda por la opción de ciudad. Al finalizar la búsqueda en la parte superior de la tabla observamos unas barras estadísticas que nos muestran la cantidad de puertos existentes en dicha ciudad.</p>
<p>Resultado deseado: Este análisis nos ayuda a observar la cantidad de puertos existentes en la ciudad donde se realiza la búsqueda.</p>
<p>Evaluación de la prueba: Se comprueba el resultado esperado. Aprobación del cliente 100%.</p>