

**RASPBERRY PI COMO DISPOSITIVO DE BORDE PARA LA DETECCIÓN DE
VULNERABILIDADES EN DISPOSITIVOS IoT**

LUIS FELIPE NARANJO HERNÁNDEZ

**FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
BOGOTÁ D.C
2021**

**RASPBERRY PI COMO DISPOSITIVO DE BORDE PARA LA DETECCIÓN DE
VULNERABILIDADES EN DISPOSITIVOS IoT**

LUIS FELIPE NARANJO HERNÁNDEZ

**Trabajo de Grado Presentado como Requisito para Obtener el Título de Ingeniero de
Sistemas**

Director
JAVIER DAZA PIRAGAUTA
Docente Investigador

Codirector
Jhon Edgar Castro Montaña
Docente Investigador

FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
BOGOTÁ D.C
2021

NOTA DE ACEPTACIÓN

Firma Presidente del Jurado

Firma del Jurado

Firma del Jurado

07 de diciembre de 2021

RESUMEN

El uso constante de tecnologías en el día a día de las personas, ha trascendido su utilización en el desarrollo de las tareas cotidianas y el impacto que conlleva la ejecución de cada una de las mismas, en donde cada aparato tecnológico cumple una o varias funciones que facilitan la vida a la humanidad, dando la capacidad en el ahorro de tiempo, liberando la carga de actividades repetitivas y aprovechadas en uso del tiempo en actividades productivas y de mayor valor.

Por consiguiente, entre estas tecnologías disponibles se encuentra IoT (Internet de las cosas) en el cual se estima que en 2021 hay 35 mil millones de dispositivos conectados a internet y cada día son integrados muchos más a nivel mundial, por lo que se distinguen al ser objetos del uso común conectados a internet, abordando campos que van desde los empresariales hasta lo domésticos, garantizando así mayor eficiencia, productividad e innovación. Sin embargo, el constante flujo de datos e información recolectada, procesada y enviada entre dispositivos es abrumador, siendo de gran atractivo por ciberdelincuentes que buscan robar información confidencial, secuestrar por medio de ransomware dispositivo y causando funcionamientos errados de los mismos.

Por tal razón, el presente proyecto se enfoca en el desarrollo/utilización de un software que automatice las pruebas a nivel de seguridad informática en estos dispositivos, permitiendo identificar vulnerabilidades en el software que puedan ser explotadas por actores externos y puedan comprometer la seguridad de una organización, trayendo consigo problemas reputacionales y económicos.

Así, el software construido permite hacer un descubrimiento, evaluación, resultado del proceso de explotación, recomendaciones de remediación, seguimiento al proceso de remediación y verificación de vulnerabilidad mitigada. Todos y cada uno de estos pilares integrado en una tarjeta Raspberry Pi, que es una computadora de tamaño reducido que se asemeja a las funciones normales de un computador convencional y que es implementado como dispositivo de borde, permitiendo un escaneo tanto intento como externo de los dispositivos IoT que se encuentran en un entorno.

Finalmente, enmarcado en pruebas sobre ambientes controlados se realiza un proceso de escaneo de vulnerabilidades a un dispositivo IoT, generando un informe acorde al nivel de exposición tanto interno/externo que permite comprender el nivel de riesgo a amenazas que podrían ser usadas para el robo de información y toma control en el comportamiento del mismo.

Palabras clave: IoT, Raspberry Pi, Vulnerabilidad, Amenaza, Superficie de Ataque, Vector de Ataque, Computación de Borde, Industria 4.0

ABSTRACT

The constant use of technologies in the daily lives of people, has transcended its use in the development of daily tasks and the impact that entails the execution of each of them, where each technological device fulfills one or more functions that facilitate life to humanity, giving the ability to save time, releasing the burden of repetitive activities and leveraged in the use of time in productive activities and higher value.

Therefore, among these available technologies is IoT (Internet of Things) in which it is estimated that in 2021 there are 35 billion devices connected to the internet and every day many more are integrated worldwide, so they are distinguished by being objects of common use connected to the internet, addressing fields ranging from business to domestic, thus ensuring greater efficiency, productivity and innovation. However, the constant flow of data and information collected, processed and sent between devices is overwhelming, being of great attraction for cybercriminals seeking to steal confidential information, hijacking through ransomware device and causing malfunctions of the same.

For this reason, the present project focuses on the development/use of a software that automates computer security tests on these devices, allowing to identify vulnerabilities in the software that can be exploited by external actors and can compromise the security of an organization, bringing with it reputational and economic problems.

Thus, the software built allows to make a discovery, evaluation, result of the exploitation process, remediation recommendations, follow-up of the remediation process and verification of the mitigated vulnerability. Each and every one of these pillars integrated into a Raspberry Pi card, which is a small-sized computer that resembles the normal functions of a conventional computer and is implemented as an edge device, allowing both attempted and external scanning of IoT devices found in an environment.

Finally, framed in tests on controlled environments, a vulnerability scanning process is performed on an IoT device, generating a report according to the level of internal/external exposure that allows understanding the level of risk to threats that could be used for information theft and takes control of the device's behavior.

Keywords: IoT, Raspberry Pi, Vulnerability, Threat, Attack Surface, Attack Vector, Edge Computing, Industry 4.0

INTRODUCCIÓN

La hiperconectividad de la industria 4.0 que cubre entornos empresariales y domésticos, es una realidad que de forma progresiva se introduce en la vida cotidiana de las personas sin distinguir edad o ubicación geográfica, cambiando la forma de pensar, interactuar y vivir del ser humano. Brindando mayor facilidad a la hora de realizar sus actividades diarias, en donde ya se habla de IoT, que dan paso a la conexión de objetos físicos a internet.

Es evidente entonces el uso de refrigeradores, bombillas, autos, hornos, cerraduras y todo aquel objeto que se pueda imaginar que se convierte en "inteligente". Disfrutando de un proceso de recolección de información, procesamiento y toma acciones en un entorno, dando la capacidad de "aprender" y modificar su código para brindar un resultado óptimo a la tarea asignada.

Ahora bien, todos estos dispositivos son producidos y comercializados en un mundo digitalmente abierto a amenazas, en donde la regularización y estandarización de requisitos mínimos en ciberseguridad. Fugas de información, ataques DDoS a gran escala, ransomware, botnets y/o malware destructivo en infraestructuras críticas se han convertido en tendencia por el auge actual. Con este proyecto de investigación se busca identificar vulnerabilidades mediante el prototipado de un software que permite detectar la superficie de ataque presentes en estos dispositivos, mostrando aspectos de mejora y soluciones que minimicen un compromiso del activo.

En síntesis, el presente proyecto tiene como bases tecnológicas el internet de las cosas (IoT) y Ciberseguridad que hacen parte de la industria 4.0, creando un hito en el desarrollo industrial que podría marcar importantes cambios sociales en los próximos años por medio de tecnología de punta, ayudando a la creación de Smart-Industries y Smart-Cities, donde es necesario contar con tecnología confiable, precisa e inteligente.

ÍNDICE DE CONTENIDO

1. CONTEXTO DE LA INVESTIGACIÓN	1
1.1 DESCRIPCIÓN DEL CONTEXTO	1
1.2 FORMULACIÓN DEL PROBLEMA.....	3
1.3 JUSTIFICACIÓN	3
1.4 OBJETIVOS	4
1.4.1 Objetivo General.....	4
1.4.2 Objetivos Específicos.....	4
1.5 ALCANCE Y LIMITACIONES	4
1.5.1 Alcance.....	4
1.5.2 Limitaciones	5
1.6 LÍNEA DE INVESTIGACIÓN	5
2 MARCO REFERENCIAL.....	7
2.1 MARCO DE ANTECEDENTES	7
2.1.1 Botnet Mirai.....	7
2.2 MARCO TEÓRICO	10
2.2.1 Placas de tamaño reducido	10
2.2.2 Sistemas Operativos para Tarjetas de Tamaño Reducido	18
2.2.3 Tecnologías de Borde	19
2.2.4 Industria 4.0	23
2.2.5 Internet of Things (IoT)	26
2.2.6 Seguridad Informática	34
2.2.7 Vulnerabilidades	35
2.2.8 Vulnerabilidades en IoT	52
2.2.9 Metodologías Ágiles	56
2.2.10 Interfaces de Programación de Aplicaciones	60
3 INGENIERÍA DEL PROYECTO	61
3.1 ASPECTOS METODOLÓGICOS DE LA INVESTIGACIÓN	61
3.1.1 Tipo de Investigación	61
3.1.2 Metodología	61
3.2 DISEÑO Y DESARROLLO	63
3.2.1 Preparación del Proyecto	63

3.2.2	Product Backlog	63
3.2.3	Análisis de Requerimientos	72
3.2.4	Desarrollo de Casos de Uso	75
3.2.5	Diagrama de Clases.....	93
3.2.6	Diagrama de Componentes	93
3.2.8	Diagrama Relacional	95
3.2.9	Estructura API REST	97
3.2.10	Arquitectura de Software	98
3.2.11	Sprint Planning	98
3.2.12	Características y Funcionalidades del Software	129
4.	PRUEBAS Y RESULTADOS	132
4.1	ALCANCE DE PRUEBAS	132
4.1.1	Resumen de Pruebas	132
4.2	ENTORNO Y CONFIGURACIÓN DE LAS PRUEBAS	132
4.3	CRITERIOS DE APROBACIÓN	133
4.4	ESTRATEGIA Y PLAN DE EJECUCIÓN DE PRUEBAS	133
4.5	DESARROLLO DE PRUEBAS Y RESULTADOS	134
4.5.1	Desarrollo Fase I de Pruebas	134
4.5.2	Desarrollo Fase II de Pruebas	152
4.5.3	Resultados	155
5.	CONCLUSIONES Y TRABAJOS FUTUROS	157
5.1	CONCLUSIONES	157
5.2	TRABAJOS FUTUROS	157
6.	REFERENCIAS BIBLIOGRÁFICAS	158
7.	ANEXOS	166

ÍNDICE DE TABLAS

Tabla 1. Características Placa de Tamaño Reducido Raspberry Pi	15
Tabla 2. Comparación Características Placas de Tamaño Reducido.....	17
Tabla 3. Perspectiva Bajo la Industria 4.0	24
Tabla 4. Clasificación de Sensores IoT	26
Tabla 5. Protocolos Asociados a IoT	30
Tabla 6. Bases de Datos de Vulnerabilidades	47
Tabla 7. OWASP Vulnerabilidades IoT 2018	54
Tabla 8. Objetivos de Seguridad de la Clase de Cumplimiento - IoT Security Compliance Framework	56
Tabla 9. Plan de Trabajo	63
Tabla 10. Ingreso a Aplicación Web.....	66
Tabla 11. Comprobar Primer Ingreso a Sistema.....	66
Tabla 12. Comprobar estado de Raspberry Pi	67
Tabla 13. Reconocer Exposición de Objetos IoT en Internet	67
Tabla 14. Reconocer Dispositivos IoT Activos	67
Tabla 15. Enumerar Puertos y Servicios en Objeto IoT	68
Tabla 16. Seleccionar Dispositivo IoT para Escaneo de Vulnerabilidades	68
Tabla 17. Selección de Plugins para Escaneo de Vulnerabilidades	68
Tabla 18. Iniciar Escaneo de Vulnerabilidades Sobre Objeto IoT	68
Tabla 19. Mostrar Resultado Escaneo de Vulnerabilidades de Objeto IoT	69
Tabla 20. Agendar Actividad de Remediación de Vulnerabilidad.....	69
Tabla 21. Monitorear de Manera Global el Estado de una Red	70
Tabla 22. Product Backlog	70
Tabla 23. Requerimientos Funcionales.....	73
Tabla 24. Requerimientos No Funcionales.....	74
Tabla 25. Caso de Uso General	75
Tabla 26. Caso de Uso Autenticación de Usuario	77
Tabla 27. Caso de Uso Cambio de Credenciales por Defecto	77
Tabla 28. Caso de Uso Estado Dispositivo Raspberry Pi	78
Tabla 29. Caso de Uso Consultas Shodan.....	79
Tabla 30. Caso de Uso para Crear Escaneo de Descubrimiento de Objetos IoT	80
Tabla 31. Caso de Uso para Modificar Escaneo de Descubrimiento de Objetos IoT	81
Tabla 32. Caso de Uso para Eliminar Escaneo de Descubrimiento de Objetos IoT	82
Tabla 33. Caso de Uso para Iniciar Escaneo de Descubrimiento de Objetos IoT	83
Tabla 34. Caso de Uso para Mostrar el Resultado del Escaneo de Descubrimiento de Objetos IoT	84
Tabla 35. Caso de Uso Seleccionar Objetivos para Escaneo de Vulnerabilidades	85
Tabla 36. Caso de Uso Seleccionar Plugins para Escaneo de Vulnerabilidades.....	86
Tabla 37. Caso de Uso Iniciar Escaneo de Vulnerabilidades	87
Tabla 38. Caso de Uso para Mostrar el Resultado Escaneo de Vulnerabilidades	88
Tabla 39. Caso de Uso para Crear Actividad de Remediación	89
Tabla 40. Caso de Uso para Modificar Actividad de Remediación	90
Tabla 41. Caso de Uso para Eliminar Actividades de Remediación	91
Tabla 42. Caso de Uso Dashboard	92
Tabla 43. Serie Fibonacci Puntuación de HU	99

Tabla 44. Resumen Sprint Uno	100
Tabla 45. Programación Sprint Uno	101
Tabla 46. Resumen Sprint Dos	106
Tabla 47. Programación Sprint Dos	106
Tabla 48. Resumen Sprint Tres	108
Tabla 49. Programación Sprint Tres	109
Tabla 50. Resumen Sprint Cuatro	111
Tabla 51. Programación Sprint Cuatro	112
Tabla 52. Resumen Sprint Cinco	114
Tabla 53. Programación Sprint Cinco	115
Tabla 54. Resumen Sprint Seis	117
Tabla 55. Programación Sprint Seis	118
Tabla 56. Resumen Sprint Siete	122
Tabla 57. Programación Sprint Siete	122
Tabla 58. Resumen Sprint Ocho	124
Tabla 59. Programación Sprint Ocho	125
Tabla 60. Resumen Sprint Nueve	127
Tabla 61. Programación Sprint Nueve	128
Tabla 62. Usabilidad Entornos Privados o Expuestos en Internet	130
Tabla 63. Resumen de Pruebas	132
Tabla 64. Estrategia y Plan de Pruebas	134
Tabla 65. Muestra Amenaza ESP32	151
Tabla 66. Pruebas Unitarias de Autenticación	152
Tabla 67. Pruebas Unitarias Módulo Estado Dispositivo	152
Tabla 68. Pruebas Unitarias Módulo Shodan	152
Tabla 69. Pruebas Unitarias Módulo Gestión de IoT	153
Tabla 70. Pruebas Unitarias Módulo Scan Vulnerabilidades	153
Tabla 71. Pruebas Unitarias Módulo Gestión de IoT	153
Tabla 72. Pruebas Unitarias Módulo Dashboard	154
Tabla 73. Resumen de Pruebas	155

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Dispositivo de Borde Arquitectura IoT	3
Ilustración 2. Dispersión de la Botnet Mirai a Nivel Global	8
Ilustración 3. Línea de Tiempo Botnet Mirai.....	9
Ilustración 4. Placa de Tamaño Reducido Hummingboard Edge	10
Ilustración 5. Placa de Tamaño Reducido ODROID-C4	11
Ilustración 6. Placa de Tamaño Reducido Banana Pi M4	12
Ilustración 7. Placa de Tamaño Reducido Arduino YÚN	13
Ilustración 8. Forma y Componentes Raspberry Pi 4	15
Ilustración 9. Proyecto Onion Pi	16
Ilustración 10. Red de Borde	22
Ilustración 11. Enfoques Router de Borde	23
Ilustración 12. Industria 4.0 por Sectores	25
Ilustración 13. Arquitectura IoT	28
Ilustración 14. Tendencia IoT Año 2020	33
Ilustración 15. Triada de la Seguridad Informática	34
Ilustración 16. Histórico de Vulnerabilidades Año 2018	36
Ilustración 17. Total de Vulnerabilidades Principales por Proveedor	36
Ilustración 18. Tabla de Precios Exploits Zerodium	41
Ilustración 19. Numeración de CVE	44
Ilustración 20. CVE por Año	44
Ilustración 21. Equipos de Respuesta de Incidentes en Colombia	46
Ilustración 22. Ciclo de Vida de una Vulnerabilidad	49
Ilustración 23. Simulación Entorno IoT	53
Ilustración 24. Metodología Desarrollo de Proyecto	62
Ilustración 25. Caso de Uso General	76
Ilustración 26. Caso de Uso Autenticación de Usuario	77
Ilustración 27. Caso de Uso Cambio de Credenciales por Defecto	78
Ilustración 28. Caso de Uso Estado Dispositivo Raspberry Pi	79
Ilustración 29. Caso de Uso Consultas Shodan	80
Ilustración 30. Caso de Uso para Crear Escaneo de Descubrimiento de Objetos IoT	81
Ilustración 31. Caso de Uso para Modificar Escaneo de Descubrimiento de Objetos IoT	82
Ilustración 32. Caso de Uso para Eliminar Escaneo de Descubrimiento de Objetos IoT ...	83
Ilustración 33. Caso de Uso para Iniciar Escaneo de Descubrimiento de Objetos IoT	84
Ilustración 34. Caso de Uso para Mostrar el Resultado del Escaneo de Descubrimiento de Objetos IoT	85
Ilustración 35. Caso de Uso Seleccionar Objetivos para Escaneo de Vulnerabilidades	86
Ilustración 36. Caso de Uso Seleccionar Plugins para Escaneo de Vulnerabilidades	87
Ilustración 37. Caso de Uso para Iniciar Escaneo de Vulnerabilidades	88
Ilustración 38. Caso de Uso para Mostrar el Resultado Escaneo de Vulnerabilidades	89
Ilustración 39. Caso de Uso para para Crear Actividad de Remediación	90
Ilustración 40. Caso de Uso para Modificar Actividad de Remediación	91
Ilustración 41. Caso de Uso para Eliminar Actividades de Remediación	92
Ilustración 42. Caso de Uso Dashboard	92
Ilustración 43. Diagrama de Clases	93
Ilustración 44. Diagrama de Componentes	94

Ilustración 45. Diagrama de Despliegue	95
Ilustración 46. Diagrama Relacional	96
Ilustración 47. Interacción de Aplicación y BBDD	97
Ilustración 48. Estructura API	97
Ilustración 49. Arquitectura por Capas	98
Ilustración 50. Cronograma Planning Uno	100
Ilustración 51. Mockup Login de Usuario	102
Ilustración 52. Mockup Cambio de Credenciales por Defecto	102
Ilustración 53. Half-Open Scan y Estado de Dispositivo.....	104
Ilustración 54. Cronograma Planning Dos.....	105
Ilustración 55. Mockup Módulo de Gestión de IoT	107
Ilustración 56. Cronograma Planning Tres.....	107
Ilustración 57. Escaneo de Vulnerabilidades en Objetos IoT	110
Ilustración 58. Cronograma Planning Cuatro	110
Ilustración 59. Mockup Módulo Scan de Vulnerabilidades	112
Ilustración 60. Cronograma Planning Cinco.....	113
Ilustración 61. Mockup Resultado de Vulnerabilidades	115
Ilustración 62. Cronograma Planning Seis	116
Ilustración 63. Gestión de Ciclo de Vida de Vulnerabilidad	120
Ilustración 64. Cronograma Planning Siete.....	121
Ilustración 65. Mockup Gestión de Vulnerabilidades	123
Ilustración 66. Cronograma Planning Ocho	123
Ilustración 67. Mockup Dashboard.....	125
Ilustración 68. Cronograma Planning	126
Ilustración 69. Mockup Estado Dispositivo.....	128
Ilustración 70. Mockup Shodan	129
Ilustración 71. Clasificación Control de Calidad de Software	133
Ilustración 72. Topología Pruebas en Entorno Privado	135
Ilustración 73. Simulación de Objeto IoT I.....	135
Ilustración 74. Simulación de Objeto IoT II.....	136
Ilustración 75. Creación de Escaneo IoT ESP32	136
Ilustración 76. Inicio de Proceso de Descubrimiento	137
Ilustración 77. Resultado Proceso de Escaneo ESP32	137
Ilustración 78. Selección de Objetivo Para Detección de Vulnerabilidades	138
Ilustración 79. Selección de Plugins Para Descubrimiento de Vulnerabilidades	138
Ilustración 80. Inicio Proceso de Detección de Vulnerabilidades	139
Ilustración 81. Resultado Escaneo de Vulnerabilidades	140
Ilustración 82. Resumen Hallazgos Encontrados	140
Ilustración 83. Resultado HTTP METHODS ESP32	141
Ilustración 84. Resultado HTTP HEADER ESP32	142
Ilustración 85. Resultado HTTP SITEMAP ESP32.....	142
Ilustración 86. Resultado Protocolo Vulnerable	142
Ilustración 87. Resultado HTTP PHPMYADMIN ESP32	143
Ilustración 88. Explotación de Vulnerabilidad Directory Traversal ESP32	144
Ilustración 89. Topología Pruebas en Entorno Público	145
Ilustración 90. Simulación de Objeto IoT I.....	145
Ilustración 91. Simulación de Objeto IoT II.....	146
Ilustración 92. Creación de Escaneo IoT ESP32	146
Ilustración 93. Inicio de Proceso de Descubrimiento	146

Ilustración 94. Resultado Proceso de Escaneo ESP32	147
Ilustración 95. Selección de Objetivo Para Detección de Vulnerabilidades.....	148
Ilustración 96. Selección de Plugins Para Descubrimiento de Vulnerabilidades	148
Ilustración 97. Inicio Proceso de Detección de Vulnerabilidades	149
Ilustración 98. Resultado Escaneo de Vulnerabilidades	150
Ilustración 99. Resumen Hallazgos Encontrados	150
Ilustración 100. Prueba de Integración - Estrategia Top Down.....	155

1. CONTEXTO DE LA INVESTIGACIÓN

1.1 DESCRIPCIÓN DEL CONTEXTO

En la actualidad, todas las organizaciones sin importar su dimensión se enfrentan a una gran cantidad de riesgos e inseguridades ciberneticas procedentes de diversas fuentes tecnológicas y humanas que se encuentran ligadas a amenazas que explotan una amplia tipología de vulnerabilidades, como las que están presentes en las aplicaciones de software y hardware.

Como consecuencia, cada organización aplica sus propias políticas, amoldadas a su sector económico y/o necesidades, para la protección de sus sistemas de información, pero que no son actualizadas o se miden el nivel de efectividad, dejando atrás aspectos fundamentales como la conexión de dispositivos del común a sus redes, haciendo que este tipo de dispositivos se convierta en una necesidad imprescindible dentro del desarrollo y sobre todo en la generación de controles que permitan disminuir el riesgo (Bevan, 1997).

Ante la situación planteada, al estar estos objetos conectados sin esquemas de seguridad bien sea por su perspectiva de uso sencillo, aumenta las vulnerabilidades, ya que estos dispositivos y sus credenciales en la red pueden ser suplantados, comprometiendo a la organización o personas, debido a su facilidad de accesibilidad y conectividad, que descuidan aspectos notorios como la privacidad.

De forma general, los ataques informáticos buscan información del individuo o control sobre sistemas, por lo que el auge del Internet de las Cosas (Internet of Things – IoT) aumenta la búsqueda de credenciales sobre un objeto para obtener la información personal y suficiente para manipular un proceso e introducir malware. Debido a varios problemas que afectan a las plataformas informáticas tradicionales y que se han trasladado al espacio móvil con la llegada de nuevos dispositivos y sistemas operativos, que cuentan con un nivel de seguridad totalmente nuevo o muy básico (Abawajy, Shamsul, Shaila, Mohammad, & Ahmad, 2018).

De acuerdo con un estudio realizado por HP (Hewlett-Packard) a 10 dispositivos IoT de seguridad para el hogar y a sus componentes de aplicación en la nube y móviles, reveló que la totalidad de estos dispositivos contenían vulnerabilidades significativas con problemas tales como; seguridad, cifrado y autenticación (Packard, 2014). Los dispositivos seleccionados para las pruebas fueron de fabricantes líderes que operan con distintos sistemas operativos, lo que le da un mayor valor al estudio.

Dadas las condiciones que anteceden, se estima que para el año 2021 el delito cibernético le costará al mundo 6 billones de dólares anuales, frente a los 3 billones de dólares del 2015, cada vez más personas se conectan a internet, haciendo que las amenazas se vuelven más complejas (Morgan, 2019). Por otra parte, el 77% de las empresas admite que no cuenta con un plan de respuesta a incidentes de ciberseguridad que se aplique sistemáticamente en la organización dejando en evidencia la falta de concientización en este aspecto (Resilient, 2018).

Con el constante crecimiento tecnológico, se muestra a continuación un breve panorama de algunas cifras que describen el síntoma general de esta tendencia actual:

- La primera página web se publicó en 1991. Hoy en día hay más de 1.200 millones de páginas web.
- Habrá 20 mil millones de usuarios de Internet para 2022 (75 por ciento de la población mundial proyectada de 8 mil millones) y más de 7,5 mil millones de usuarios de Internet para 2030 (90 por ciento de la población mundial proyectada de 8,5 mil millones, 6 años y más viejo).
- A partir de 2025, el número de dispositivos conectados se duplicará y las personas interactuarán con un dispositivo cada 18 segundos.
- Para 2023, se espera que el mercado mundial de ciberseguridad crezca cerca de 250.000 millones de dólares.
- Más de 500 millones de dispositivos portátiles serán vendidos en todo el mundo para el año 2021. La lista de estos dispositivos portátiles incluye relojes inteligentes, monitores montados en la cabeza, cámaras portátiles, auriculares con Bluetooth y monitores de acondicionamiento físico, entre mucho otros (Magazine, 2019).

Teniendo en cuenta lo anterior y dado lo nuevo de la seguridad en objetos conectados a internet, el proyecto IoT de OWASP (Open Web Application Security Project), ha diseñado un marco para ayudar a fabricantes, desarrolladores y consumidores a comprender mejor los problemas de seguridad asociados con el IoT, y para permitir a los usuarios en cualquier contexto tomar mejores decisiones de seguridad al crear, implementar o evaluar tecnologías de IoT (OWASP, OWASP, 2018). La lista de los riesgos que conforman el manifiesto se muestra a continuación en su última versión 2018:

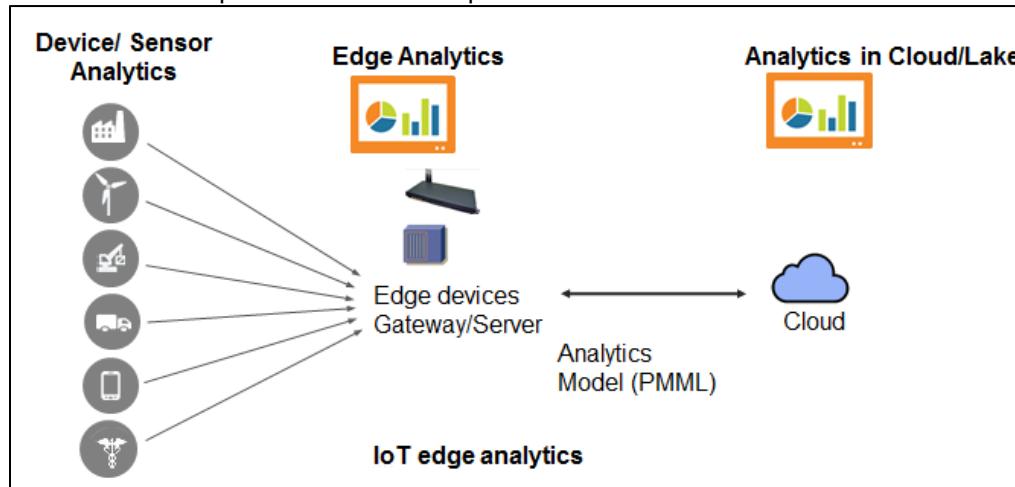
1. Contraseñas débiles, adivinables o codificadas
2. Servicios de red inseguros
3. Interfaces inseguras del ecosistema
4. Falta de mecanismo de actualización segura
5. Uso de componentes inseguros u obsoletos
6. Protección de privacidad insuficiente
7. Transferencia y almacenamiento de datos inseguros
8. Falta de gestión de dispositivos
9. Configuración predeterminada insegura
10. Falta de endurecimiento físico

Existe una relación estrecha entre los dispositivos IoT y las vulnerabilidades, debido a que la ciberseguridad está diseñada y actualizada según una necesidad base para dar cuenta de amenazas conocidas y tiene como propósito principal la prevención de infracciones de seguridad y mitigación de amenazas (internas o externas), y algunos de los sistemas están diseñados para que cuente con un sistema de protección y resistencia ante las amenazas desconocidas, encontrando un proceso de recuperación en curso después de un ataque informático que hace parte de la resiliencia en un sistema.

Teniendo presente lo anterior, la resiliencia es una medida utilizada en natural, organizacional, social y en sistemas diseñados que evalúan la capacidad de un sistema para “prepararse, recuperarse o más potencialmente adaptarse a posibles eventos adversos” (Packard, 2014). De igual forma la resiliencia está estrechamente relacionada con nociones como riesgo y vulnerabilidad, el modelado y la simulación son técnicas que se utilizan ampliamente para evaluar el riesgo, la vulnerabilidad y la resistencia de varios sistemas a los enfoques de modelado y simulación (OWASP, OWASP, 2018).

Debido al crecimiento de las redes de sensores para medir el nivel de temperatura, humedad, electromagnetismo, gases, etc., las comunicaciones Maquina a Maquina (Machine to Machine - M2M) como ejemplo; cadena de abastecimiento autosustentable, vehículos autosuficientes y servicios en IoT para la seguridad en el hogar, detección de enfermedades, gestión de suministros, se toma como ejemplo un proceso de analítica de datos (ver Ilustración 1) para resaltar que en estos sistemas y modelos cuentan con un componente común, que son los dispositivos de borde.

Ilustración 1. Dispositivo de Borde Arquitectura IoT



Fuente: Tomado de <https://bit.ly/2xtRKTJ>

Según el portal de noticias llamado Decidio, en el año 2020 hubo más de 50,000 millones de dispositivos conectados a las redes, por lo que el futuro de la computación se encuentra en el borde de la red. Algunos ejemplos son las ciudades, fábricas y casas/edificios inteligentes, la distribución de contenido de alta definición, la informática de alto rendimiento, conectividad limitada, realidad virtual, así como la digitalización de petróleo y gas (Vertiv, 2018).

1.2 FORMULACIÓN DEL PROBLEMA

¿Qué tan expuestos a ciberamenazas se encuentran objetos IoT en internet y entornos privados tomando muestras bajo pruebas en entornos controlados?

1.3 JUSTIFICACIÓN

El avanzado crecimiento de tecnologías emergentes disponibles de forma global tanto para uso común como para uso industrial, presentan la necesidad de abordar aspectos de ciberseguridad cada vez más sofisticados dado su uso frecuente, en donde especialmente el presente proyecto se enfoca en aspectos de ciberseguridad en IoT.

Estándares regulatorios y de control destinados a la seguridad de la información en entornos organizacionales tales como la ISO/IEC 27001, COBIT, PCI DSS, NIST Cybersecurity Framework, etc., concuerdan que la implementación adecuada de un programa de gestión vulnerabilidad puede reducir las fugas de información y comportamientos erróneos en activos críticos.

Adicionalmente a lo anterior, la implementación exitosa de una herramienta de software de seguridad que ayude a los cumplimientos normativos, se basa en la facilidad de despliegue y no alteración del desarrollo normal de actividades propias del negocio, lo

que garantiza la continuidad segura de la operación. Por lo que, el despliegue de la herramienta desarrollada en una Raspberry Pi, permite un desempeño óptimo, así como realizar pruebas intrusivas de seguridad en un ambiente seguro y controlado.

En ese mismo sentido, el uso intuitivo por medio de una interfaz gráfica desplegada de forma web del software, permite seguir un consecutivo de pasos para el control del ciclo de vida de una vulnerabilidad de forma fácil y amigable, ayudando tanto a personal con conocimientos técnicos como a entusiastas de la ciberseguridad a entender de forma global la superficie y vector de ataque al que son propensos los dispositivos IoT que se encuentran desplegados en una infraestructura.

En el orden de las ideas anteriores, el desarrollo del prototipo de software del presente proyecto que realiza la recolección de información pasiva y activa, análisis, muestra de resultados y seguimiento de remediación, se destina a ser una herramienta colaborativa para la gestión adecuada del ciclo de vida de las vulnerabilidades presentes en objetos IoT, ayudar al cumplimiento normativo y a la toma de decisiones de forma acertada.

1.4 OBJETIVOS

1.4.1 Objetivo General

Construir un prototipo de software para la detección de vulnerabilidades en objetos IoT usando un Raspberry Pi como dispositivo de borde

1.4.2 Objetivos Específicos

- Identificar las vulnerabilidades que se presentan comúnmente en objetos IoT y estrategias de mitigación
- Diseñar un modelo de proceso que permita la identificación y clasificación de las vulnerabilidades presentes en objetos IoT
- Desarrollar el prototipo de software para la detección de vulnerabilidad en objetos IoT con base en el modelo determinado.
- Validar el prototipo de software diseñando y aplicando pruebas controladas de ataques a objetos IoT en entornos privados e internet sirviendo como muestra para determinar el nivel de exposición de amenazas cibernéticas

1.5 ALCANCE Y LIMITACIONES

1.5.1 Alcance

El presente proyecto de investigación tiene como alcance crear un prototipo de software implementado en Raspberry Pi, basado fundamentalmente en el ya mencionado OWASP Internet of Things y haciendo uso de las mejores prácticas a la hora de mitigación de amenazas en dispositivos IoT.

Para constatar el funcionamiento final del software que permite hacer la detección de vulnerabilidades, se realiza el despliegue a nivel de infraestructura de red la tarjeta Raspberry Pi como un dispositivo de borde, analizando un objeto IoT programado de forma previa en una tarjeta ESP32 para el control una bombilla de forma remota, que se expondrá de manera interna y externa, en donde la diferente información recolectada y

detectada, funcionará como muestra base para identificar el nivel de exposición a amenazas cibernéticas sobre estos objetos conectados a internet.

1.5.2 Limitaciones

Las presentes limitaciones restringirán la investigación:

- La adquisición de diversos objetos IoT que permita diversificar y ampliar la perspectiva a amenazas expuestas que pueden ser detectadas por el prototipo de software en ambientes controlados, dada la inmensa variedad de marcas comerciales, precios, entornos de usabilidad, procesos de implementación y facilidad de configuración. Limitándose al uso de la tarjeta ESP32.
- En su mayoría, cada empresa fabricante de dispositivos IoT crea su propio protocolo de comunicación y control de sus productos, por lo que se mantienen estos bajo código cerrado, dificultando su implementación en el prototipo de software para el proceso de análisis de vulnerabilidades, así pues, se han acoplado protocolos de uso amplio como SSH, HTTP, HTTPS, Telnet, SMTP, NTP, VNC, IRC y UPnP.
- La publicación de fragmentos de software de forma libre que permiten aprovechar un agujero de seguridad conocidos como exploits, que en este caso se encuentran destinados a comprometer dispositivos IoT son muy bajos, por lo que se limita a la implementación en el prototipo de software a los encontrados en la web superficial, teniendo en cuenta que el presente proyecto no se enfocado en la generación de conocimiento en este aspecto, además el grado de dificultad y tiempo que este proceso conlleva.
- Partiendo de la Ley 1273 del 2009 instaurada en territorio colombiano denomina "de la protección de la información y de los datos" , en donde se preservan integralmente los sistemas que utilicen las tecnologías de la información y las comunicaciones (Colombia, 2009), el presente proyecto se desarrolla bajo marcos legales, limitado a hacer pruebas de vulnerabilidades bajo software intrusivo en ambientes controlados, contando con el consentimiento del administrador de la infraestructura informática.

1.6 LÍNEA DE INVESTIGACIÓN

El presente proyecto se enmarca dentro de la línea de investigación institucional GUIAS (Grupo de Investigación Aplicada en Señales y Sistemas) (Libertadores, 2019) enfocado bajo el marco de ciberseguridad dedicado a implementar soluciones y procesos que aseguren la información tanto de amenazas externas como internas, desde su origen hasta su aprovechamiento para las operaciones de las empresas y para la toma de decisiones.

Teniendo en cuenta algunas medidas que impulsen día a día soluciones dirigidas a incrementar la seguridad de la información como; el crecimiento de las redes sociales, los dispositivos móviles, el correo electrónico, la Big Data, entre otras. Para ello, se diseñarán estrategias y se conocerán herramientas para trabajar entornos seguros, dirigidos a garantizar la continuidad de las operaciones y permanencia en los mercados de las empresas.

Contando con los conocimientos adquiridos en el semillero de Investigación en Resiliencia Cibernética y el semillero de Investigación Seguridad Informática el objetivo que se plantea por medio de estas líneas, es llegar por medio de la generación de

conocimiento de todos los elementos desarrollados por medio de esta investigación, en dar a conocer las ciberamenazas a los que se encuentran expuestos objetos IoT en un mundo globalmente conectado, para atender la necesidad de seguridad en estos sistemas y poder afrontarse de forma individual o a nivel organización.

2 MARCO REFERENCIAL

2.1 MARCO DE ANTECEDENTES

2.1.1 Botnet Mirai

En el mes de octubre del año 2016, parte de internet a nivel global sucumbió frente al ataque de DDOS (Distributed Denial of Service) más grande y potente registrado hasta esa fecha (S2Grupo, 2017), comprendiendo un alrededor de 1,2 Tbps de tráfico (Calvo Ortega & García Valdés, 2018), en donde la compañía Dyn (Dynamic Network Services, Inc) - quien es el proveedor líder de rendimiento de Internet para las webs más visitadas del mundo, según lo medido por Alexa 1000, incluidos Twitter, Netflix, Pandora, Zappos, CNBC, Etsy, BT, Hershey, The Guardian y Seeking Alpha (ICANNWiki, 2019) - fue atacada por medio de una botnet denominada “Mirai”, compuesta por su mayoría de dispositivos de toda clase de IoT.

Aunque Mirai no fue la primera de su especie (Gafgyt o Remaiten, entre otras, habían llegado antes), sí que definió un antes y un después en cuanto a la comprensión del gran público que suponen las amenazas de los dispositivos IoT, teniendo en cuenta la dimensión global que abarca y al amplio público al que puede llegar (S2Grupo, 2017).

De acuerdo con Kaspersky, en su artículo denominado “Mirai: The fall of the Internet”, los dispositivos IoT ciertamente les dio nueva vida a las botnet, en donde estos dispositivos la seguridad jamás se había estimado ni diseñado y para los cuales no existía ningún antivirus, comenzaron a infectarse esporádicamente a gran escala. Rápidamente, estos dispositivos rastrearon otros del mismo tipo y transmitieron el contagio de inmediato (Kochetkova, 2016).

Tal como se observa, el fileless malware - denominado así porque permanece en memoria RAM sin dejar rastro en el disco - hace uso de la gran vulnerabilidad que presentan diversos dispositivos IoT, la implementación del protocolo UPnP, siendo este el permite el acceso desde una red externa hasta el dispositivo a través del NAT (Network Address Translation) de la red, de esta forma se publica el puerto de acceso a todo Internet (S2Grupo, 2017). Este aspecto, junto a la configuración en su mayoría de las credenciales por defecto y el hecho de que el usuario no es consciente de la infección han permitido a la botnet crecer.

De forma general, se descubrió 49,657 direcciones IP únicas que albergaban dispositivos infectados por Mirai, las direcciones IP de los dispositivos infectados con Mirai se detectaron en 164 países.

Como lo muestra la Ilustración 2, las IP de botnet están muy dispersas, apareciendo incluso en lugares remotos como Montenegro, Tayikistán, Somalia y Colombia (BenlgaDima, 2016), el cual este último se encuentra en la posición 10 del ranking a nivel global de países infectados, presentando un panorama preocupante frente a la seguridad en dispositivos IoT que utilizan las personas de este país, siendo base que en muchas oportunidades estos dispositivo son tratados de forma cotidiana, importando la funcionalidad más no la seguridad de la información y la privacidad.

Ilustración 2. Dispersión de la Botnet Mirai a Nivel Global



Fuente: Tomado de <https://bit.ly/2Yu4lvU>

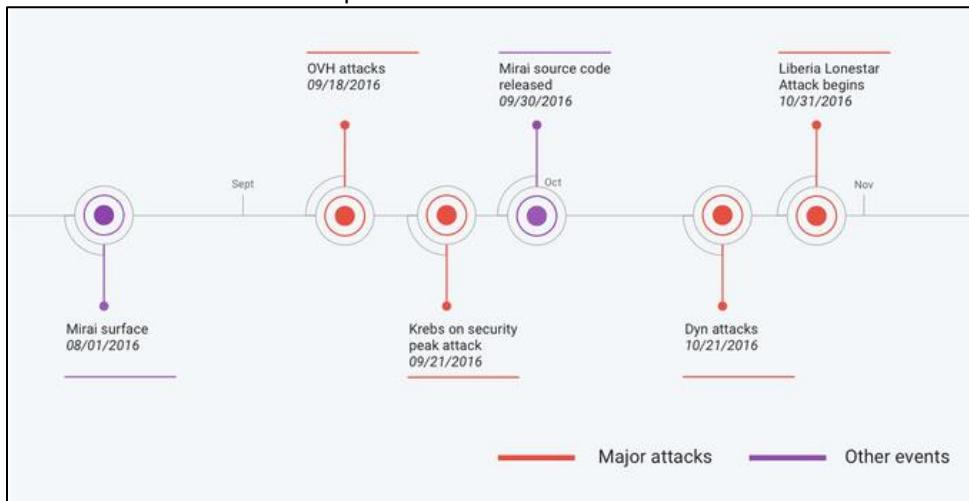
Aunque inicialmente los registros de Mirai parten del 1 de agosto de 2016, las primeras observaciones lo catalogan como una variante de Linux.DDoS. No será catalogado como un malware opuesto a Gafgyt hasta el artículo de Malware Must Die el día 31 de agosto. Tras más de mes y medio de propagación de la botnet, se produjo el primer ataque. Más de 600 Gbps fueron emitidos contra el portal krebsonsecurity.com el día 20 de septiembre. Pocos días después OVH reportó haber sufrido ataques de más de 1 Tbps (S2Grupo, 2017).

Mientras transcurria el pasar del tiempo, el apoderamiento de dispositivos IoT de esta botnet dio un vuelco insospechado ocurrió el día 30 de septiembre de 2016, cuando un usuario denominado Anna-Senpai compartió el código fuente de Mirai en la plataforma Hackforums, convirtiéndose así en la primera botnet IoT Open Source.

Finalmente, el día 21 de octubre de 2016 como se muestra en la Ilustración 3, una serie de ataques DDoS a través de Internet de las cosas contra la infraestructura de DNS gestionada por Dyn impidió el acceso de los usuarios a los sitios web

más importantes de EE. UU, entre los que se encontraban Twitter, Spotify, PayPal, Amazon, GitHub, Starbucks y un alrededor de 60 páginas más (Akamai, 2016). Probando de esta forma que aparte de los ataques dirigidos, las organizaciones también deben gestionar el riesgo de sufrir ataques DDoS contra la infraestructura central de Internet, por ejemplo, los servidores DNS.

Ilustración 3. Línea de Tiempo Botnet Mirai



Fuente: Tomado de <https://bit.ly/2L1WaVa>

En este punto es importante entender cuál es el papel que cumple Dyn y su infraestructura DNS, en el cual el Domain Name System (DNS) o sistema de nombres de dominio es el sistema que conecta un navegador (Google Chrome, Firefox, Internet Explorer, Safari, etc.) con el sitio web que está buscando.

En este propósito, cada sitio tiene una dirección digital denominada dirección IP, así como una URL más entendible y acorde al sitio que se visita. Por ejemplo, blog.ejemplo.com vive en la dirección IP 154.34.68.12. Básicamente, un servidor DNS funciona como una libreta de direcciones: le dice a un navegador en qué ubicación digital está almacenado un sitio. Si un servidor DNS no responde a una solicitud, el navegador no sabrá cómo cargar la página (Kochetkova, 2016). Por esta razón los proveedores de DNS (especialmente los principales) forman una parte importante de la infraestructura crítica de Internet.

De igual forma, un ataque de denegación de servicio distribuido (DDoS) inunda los servidores que ejecutan un sitio web o un servicio en línea con solicitudes hasta que colapsan y los sitios a los que sirven dejan de funcionar, como en el caso de DNS. Para llevar a cabo un ataque DDoS, los delincuentes deben enviar una cantidad considerable de solicitudes, siendo así necesario muchos dispositivos para hacerlo. Para un ataque DDoS, generalmente usan ejércitos de computadoras pirateadas, teléfonos inteligentes, dispositivos y otras cosas conectadas (Kochetkova, 2016). Trabajando juntos (pero sin el conocimiento o consentimiento de sus propietarios) estos dispositivos forman botnets.

2.2 MARCO TEÓRICO

2.2.1 Placas de tamaño reducido

Las placas de tamaño reducido o conocidas también como Single Board Computers (SBCs) son descritos como una computadora completa construida en una placa de circuito único, con microprocesador, memoria, Input/Output (I/O) y otras características requeridas de la computadora tradicional (Steven J. Johnston, 2018).

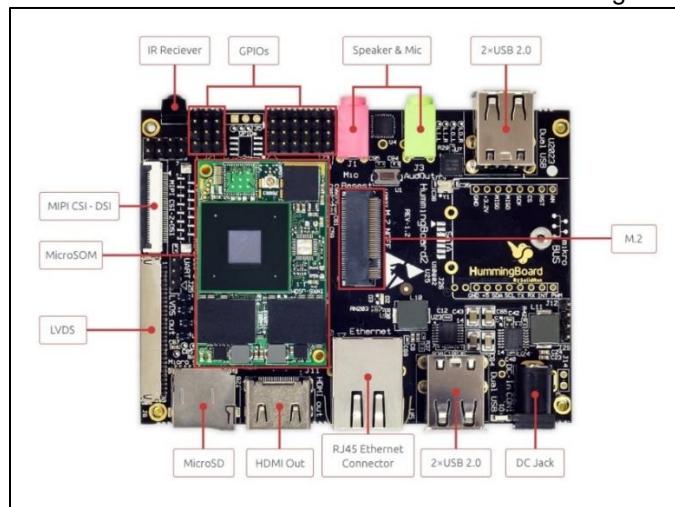
En la actualidad estas placas son lo suficientemente poderosas que pueden ejecutar diferentes sistemas operativos y cargas de trabajo de flujo principal. Muchos de estos tableros pueden vincularse entre sí (Steven J. Johnston, 2018), para crear pequeños grupos de bajo costo que replican características de grandes data centers, y que pueden habilitar nuevas aplicaciones como Fog Computing o Edge Computing donde la computación se expulsa del núcleo de la red hacia las fuentes de datos, por ello, algunas placas de tamaño reducido referentes en el mercado y tenidas en cuenta para efectos de este proyecto son:

2.2.1.1 *Hummingboard*

La Hummingboard es un dispositivo de dimensiones reducidas - como se observa en la Ilustración 4 - que es comercializado por la empresa israelí SolidRun, quien agrega todos los componentes necesarios para hacer de este un comprador de pequeño formato y bajo coste.

En ese mismo sentido, este dispositivo se encuentra equipado con un microprocesador, memoria RAM, varios puertos de entrada/salida y un lector de tarjetas SD para almacenamiento del sistema operativo o elementos digitales (Perez, 2014). Sin embargo, incorpora una ventaja que le da un valor agregado del resto de placas de tamaño reducido disponibles en el mercado y es que gracias a que permite ampliar el procesador y la memoria RAM, se pueden crear mini-computador de componentes más completos con capacidad de albergar algunas variantes de Linux.

Ilustración 4. Placa de Tamaño Reducido Hummingboard Edge



Fuente: Tomado de <https://bit.ly/2Wvm89Y>

En la anterior imagen se observa el modelo Hummingboard Edge que es una placa ideal para proyecto M2M, fue desarrollada para satisfacer la demanda del mercado B2B de un dispositivo de alto rendimiento, confiable e innovador (SolidRun, 2020). El HummingBoard Edge incluye un conjunto completamente nuevo de características como eMMC (8GB), mPCIE (tamaño medio o completo), entrada de amplio rango (7V-36V), carcasa metálica opcional, entre muchas características que la hacen adecuada para trabajos de alta demanda y entornos IoT.

2.2.1.2 ODROID

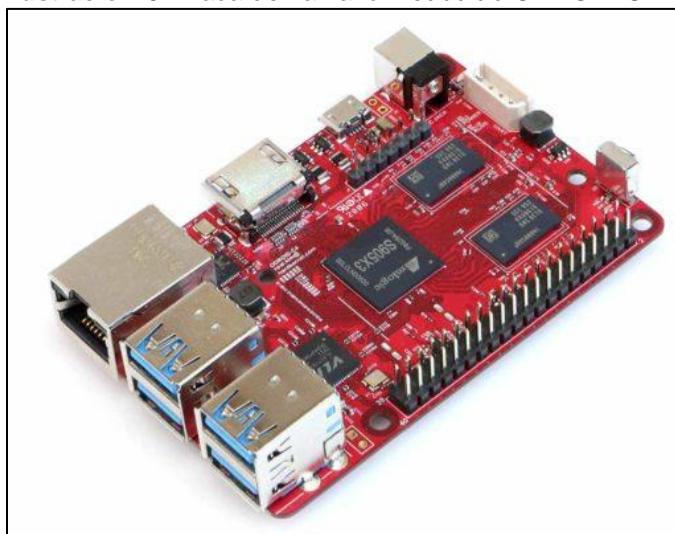
ODROID es una placa de tamaño reducido creada por una compañía coreana de hardware de código abierto llamada Hardkernel. ODROID significa Open + Droid. Es una plataforma de desarrollo para el hardware, así como el software (ODROID, Wiki ODROID, 2020).

La placa ODROID tiene una versión para desarrolladores y una versión completa. La versión de desarrollador está destinada a aquellos interesados en desarrollar aplicaciones, juegos o contenido para usuarios de Odroid (Techopedia, n.d.).

La unidad Odroid incluye una placa de depuración, códigos fuente y esquemas para ayudar a los desarrolladores. También hay una comunidad de desarrolladores de Odroid, que ayuda a promover la interacción global entre los desarrolladores y usuarios de Odroid, tanto así que se encuentran disponibles los sistemas operativos Android, Arch Linux, Fedora, Kali Linux, NetBSD, entre otros 10 sistemas operativos adicionales que lo convierten en una placa ideal para implementación en diferentes entornos.

En la actualidad la última versión de ODROID es la ODROID-C4 que se muestra en la Ilustración 5, correspondiente a una computadora de placa única de nueva generación que tiene más eficiencia energética y un rendimiento más rápido que ODROID-C2 (versión anterior del año 2016), la CPU principal del ODROID-C4 está construida con un clúster Cortex-A55 de cuatro núcleos con una GPU Mali-G31 de nueva generación (ODROID, ODROID, n.d.). Los núcleos A55 funcionan a 2.0Ghz sin estrangulamiento térmico utilizando el disipador de calor de serie que permite una computadora robusta y silenciosa.

Ilustración 5. Placa de Tamaño Reducido ODROID-C4



Fuente: Tomado de <https://bit.ly/35VC8W1>

2.2.1.3 Banana Pi

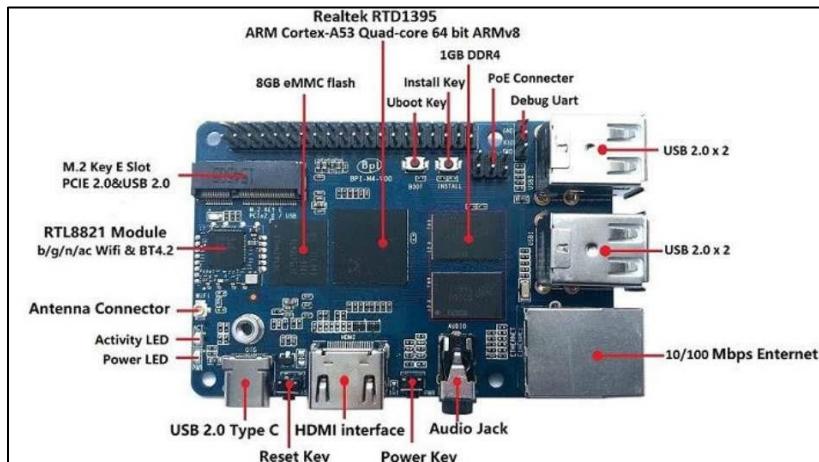
Banana Pi es un proyecto de hardware de código abierto liderado por GuangDong BiPai technology co., LTD. Se centra en una placa de desarrollo de hardware de código abierto de las series ARM y MCU, proporciona una plataforma de software y hardware abierta y crea la plataforma de desarrollo de tecnología básica (Pi B. , 2020). Serie completa de productos de hardware de código abierto, integración completa de voz, datos, plataforma de sistema de video.

Con referencia a lo anterior, los desarrolladores pueden construir de manera flexible varias plataformas de aplicaciones en la plataforma de base de hardware de código abierto. Se puede aplicar en Internet de las cosas (IoT), inteligencia artificial, control industrial de Internet, educación STEAM y otros aspectos.

En la actualidad Banana Pi cuenta con la versión Banana Pi M4 que se puede apreciar en la Ilustración 6, la cual permite la instalación de los sistemas operativos embebidos Armbian, Tina Linux y Mainline Linux uboot 2019.07 quienes recomienda en primera instancia el fabricante para aprovechamiento del potencial completo de la placa y bajo de recursos que consumen (OS, 2020).

Banana Pi no se queda atrás respecto a otras placas de tamaño disponibles en el mercado y permite la instalación de sistemas operativos de mayor auge como Ubuntu, Kali Linux, CentOS, Android y al igual que Raspberry Pi la placa Banana Pi dispone de un sistema operativo propio denominado Bananian teniendo como base el OS Debian optimizado (Elliot, 2018).

Ilustración 6. Placa de Tamaño Reducido Banana Pi M4



Fuente: Tomado de <https://bit.ly/3dMjQcv>

2.2.1.4 Arduino

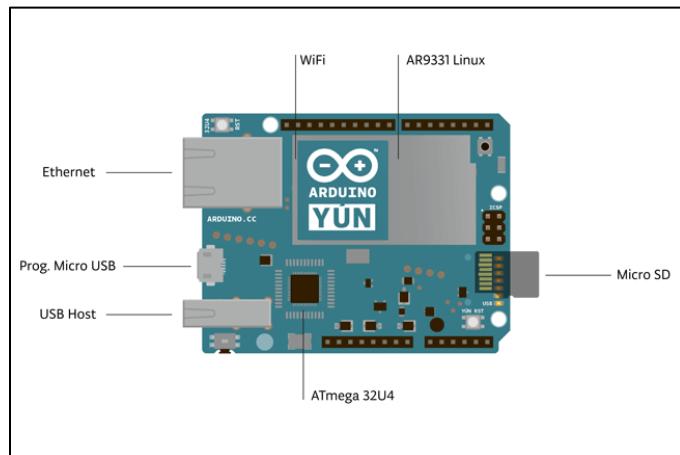
De acuerdo con la documentación oficial (ARDUINO, 2020), Arduino es una herramienta popular para el desarrollo de productos IoT, así como una de las herramientas más exitosas para la educación STEM/STEAM (Science, Technology, Engineering and Mathematics).

En efecto, cientos de miles de diseñadores, ingenieros, estudiantes, desarrolladores y fabricantes de todo el mundo están utilizando Arduino para innovar en la construcción y programación de microcontroladores que permitan la automatización de

procesos o brinden herramientas que faciliten la interacción del ser humano con el entorno tecnológico, esta placa electrónica de hardware libre (ver Ilustración 7) es aplicable a diferentes entonos como la musical, juegos, juguetes, hogares inteligentes, agricultura, vehículos autónomos, robots, drones y mucho más.

En un comienzo Arduino comenzó como un proyecto de investigación de Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino y David Mellis en el Interaction Design Institute de Ivrea a principios de la década de 2000 (ARDUINO, 2020). No obstante años más adelante, la primera placa Arduino se presentó en 2005 para ayudar a los estudiantes de diseño, que no tenían experiencia previa en electrónica o programación de microcontroladores, a crear prototipos funcionales que conectaran el mundo físico con el mundo digital. Desde entonces se ha convertido en la herramienta de creación de prototipos electrónicos más popular utilizada por ingenieros e incluso grandes corporaciones.

Ilustración 7. Placa de Tamaño Reducido Arduino YÚN



Fuente: Tomado de <https://go.aws/3dJ6nSA>

Con la programación necesaria y la ayuda de Arduino, es posible hacer que cualquier dispositivo tecnológico se haga “inteligente” (García Cobo, n.d.). Sin embargo, lo más habitual es utilizar la placa para que el gadget creado pueda conectarse a Internet y poder manipularlo a través de otro dispositivo como un smartphone, una tablet o un pc, así que por ello nació el ideal de la placa Arduino YÚN, la cual se observa en la anterior imagen, siendo este un miembro notable de la familia Arduino, unido a un minicomputador con su propia CPU (Atheros AR9331) espacio de almacenamiento (micro SD y USB) y conectividad (Ethernet, Wifi, host USB) (Cámara Nebreda, 2017).

Es importante es este punto resaltar que Arduino cuenta con memoria RAM y un procesador de características limitadas, lo cual impide ejecutar un sistema operativo completo basado en Linux como Debian, Kali Linux, Ubuntu o Arch Linux como se han visto en las anteriores tarjetas de tamaño reducido, sin embargo cuenta con una distribución basada en Linux propia denominada OpenWrt-Yun (ORG O. , 2018), este sistema operativo se ejecuta a través de una conexión IEEE 802.11 (conocida comúnmente como Wi-fi) entre la placa Arduino YÚN y un computador, siendo este último quien permite la comunicación por medio de una navegador web la administración y configuraciones de red correspondientes.

La tarjeta Arduino YÚN cumple un papel fundamental en el IoT debido a que permite realizar conexiones por medio de internet y administrar de forma remota el

encendido/apagado de un dispositivo, administración de sensores, cerraduras inteligentes, cámaras, dispositivos biométricos, etc.

Acorde a Chema Alonso (Alonso, 2018), Arduino se ha convertido en una plataforma electrónica Open Source diseñada para el uso sencillo tanto de hardware como de software, en donde el límite lo marca la imaginación, siendo una plataforma que brinda un abanico amplio de posibilidades que van desde un sistema para la apertura y cierre de la puerta de un garaje, pasando por un detector de presencia, luz y oscuridad, hasta tal punto que la seguridad informática lo ha implementado como pruebas de concepto en entornos IoT y a continuación se muestran algunos trabajos e investigaciones en donde Arduino fue una pieza fundamental en este campo:

- Arducky: Un Rubber Ducky hecho sobre Arduino para hackear Windows
- Latch y el IoT: Un timbre de puerta controlado por Latch y Arduino
- Una alcancia protegida por Latch y Biometría con Arduino
- Seguridad Criptográfica en IoT con Arduino
- Smolpion: Otro proyecto en Arduino para hacer un Rubber Ducky
- La seguridad del IoT con Edison, Arduino y Sinfonier

2.2.1.5 Raspberry Pi

Raspberry Pi es un computador de placa reducida de bajo costo siendo del tamaño de una tarjeta de crédito, diseñada y fabricada en el Reino Unido con la intención inicial de proporcionar un dispositivo informático barato para la educación. Desde su lanzamiento, sin embargo, ha crecido mucho más allá de la esfera académica.

Sus orígenes se remontan en el Laboratorio de Computación de la Universidad de Cambridge en 2006. El científico informático Eben Upton, junto con Rob Mullins, Jack Lang y Alan Mycroft, estaban preocupados de que los estudiantes de pregrado en informática se hubieran divorciado de los aspectos técnicos de la informática. Esto se debió en gran parte a los programas escolares que pusieron énfasis en el uso de las computadoras en lugar de comprenderlas.

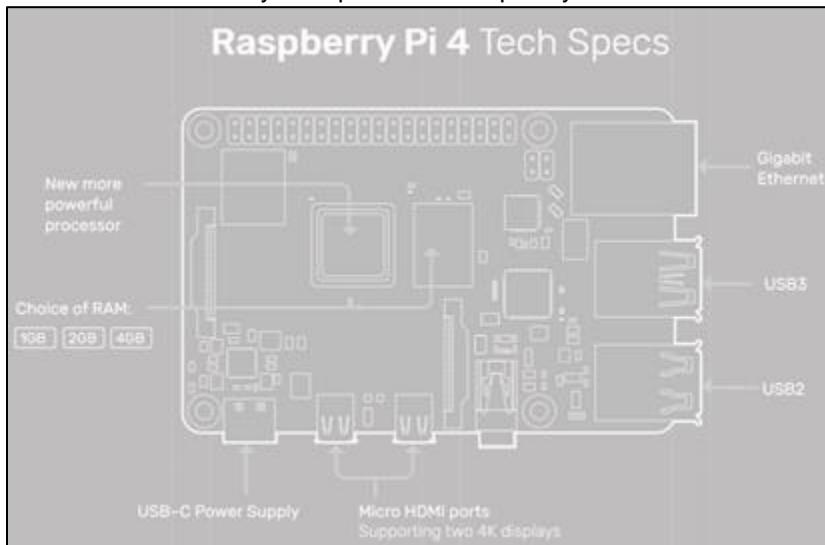
Detrás de esta preocupación inicial, se formó la fundación Raspberry Pi. Durante los próximos seis años, el equipo trabajó en el desarrollo de un dispositivo barato y accesible que ayudaría a las escuelas a enseñar conceptos como la programación, acercando así a los estudiantes a comprender cómo funciona la informática.

El lanzamiento comercial inicial de la Raspberry Pi fue en febrero de 2012. Desde entonces, ha pasado por una serie de revisiones y ha estado disponible en dos modelos, que son el Modelo A y el Modelo B.

El dispositivo Modelo A es el más económico y simple de las dos computadoras y el Modelo B el más potente, incluido el soporte para la conectividad Ethernet (Dennis, 2016).

Esta placa reducida es capaz de hacer todo lo que esperaría que hiciera una computadora de escritorio, desde navegar por Internet y reproducir videos de alta definición, hasta hacer hojas de cálculo, procesamiento de textos y jugar juegos (Pi R., Raspberry Pi, 2014). En la Ilustración 8 se puede evidenciar la forma y componentes que dispone la Raspberry Pi 4 que al momento de escribir este documento es la versión más reciente.

Ilustración 8. Forma y Componentes Raspberry Pi 4



Fuente: Tomado de <https://bit.ly/2RznL3y>

La Raspberry Pi 4 cuenta con una serie de características que la convierte de un ordenador de placa reducida a un ordenador de escritorio con especificaciones de hardware poderosas que se pueden evidenciar en la Tabla 1 que se muestra a continuación, ideales para proyectos que van desde computadores personales, robótica, consolas de videojuego, media-center, teléfonos móviles, estaciones meteorológicas, servidores web y correo, automatización de procesos hasta la integración de machine learning, el límite está en la imaginación, es un dispositivo que puede integrarse en diversos ambientes, situaciones, mercados, industrias y ambientes (Dennis, 2016).

Tabla 1. Características Placa de Tamaño Reducido Raspberry Pi

Característica	Especificaciones
Procesador	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memoria	1GB, 2GB o 4GB LPDDR4 (dependiendo del modelo)
Conectividad	2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 ports 2 × USB 2.0 ports
GPIO	Standard 40-pin GPIO header (compatible entre diferentes versiones de Raspberry Pi)
Video & Sonido	2 × micro puertos HDMI (up to 4Kp60 soportado) 2-lane MIPI DSI display port 2-lane MIPI CSI cámara port 4-pole sonido estéreo y puerto de video
Multimedia	H.265 (4Kp60 decode); H.264 (1080p60 decode, 1080p30 encode); OpenGL ES, 3.0 graphics

Tarjeta SD Soportada	Entrada Micro SD para carga de sistema operativo y guardado de información
Fuente de Alimentación	5V DC via USB-C connector (minimum 3A1) 5V DC via GPIO header (minimum 3A1) Power over Ethernet (PoE)-enabled (requires separate PoE HAT)
Ambiente	Temperatura operativa entre 0 – 50° C

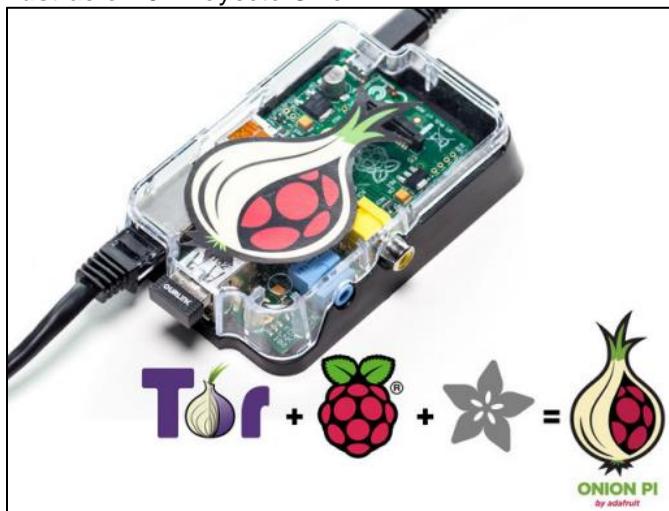
Fuente: Tomado de <https://bit.ly/2xtUEYF>

Además, Raspberry Pi tiene la capacidad de interactuar en el mundo de la seguridad informática en donde nos abre la posibilidad a la securización de entornos informáticos domésticos y privados (Lakhani & Muniz, 2015), encontrando así proyectos interesantes de alto valor que permiten mantener la privacidad, evitar robos de información, evitar espionaje por parte de ISP's y evitar censuras de acuerdo a la ubicación de un usuario determinado.

Como primer ejemplo se encuentra un proyecto denominado “Solución de VPN basa en Raspberry Pi” como trabajo final de master liderado por Álvaro Núñez, Romero Casado, Javier José Pecete García, Alejandro Amorín Niño y Juan Antonio Baeza Miralles, en este documento se describe como montar un servicio de VPN personal sobre Raspberry Pi haciendo uso del software libre OpenVPN y Latch que es una aplicación propietaria del grupo Telefónica (Álvaro Núñez-Romero Casado, 2016).

Otro proyecto con Raspberry Pi diseñado para mantener el anonimato y la privacidad denominado “Onion Pi”, proyecto desarrollado por la organización Adafruit Learning System quien es una organización dedicada a la creación de gadget implementados en sectores como la domótica con placas reducidas como Arduino, Raspberry Pi, Adafruit, circuitos embebidos, entre algunos otros (system, 2018). Con Onion Pi como se muestra en la Ilustración 9 es posible conectar un portátil, smartphone, tablet o dispositivo de con tecnología IEEE 802.11, de esta forma todo el tráfico TCP generado de alguno de estos dispositivos puede ir a través de la red Tor (The Onion Routing) brindando un encapsulamiento de 3 capas hasta su destino, permitiendo así la facilidad de anonimato.

Ilustración 9. Proyecto Onion Pi



Fuente: Tomado de <https://bit.ly/3bq8RED>

2.2.1.6 Comparación de Placas de Tamaño Reducido.

Como se ha tratado a lo largo de este apartado, cada una de las tarjetas de tamaño reducido tienen características, ventajas, desventajas y una comunidad que las convierten en una plataforma acorde a la necesidad o tipo de proyecto que esté trabajando, por ello a continuación, en la Tabla 2 se muestra una comparación de características de las diferentes tarjetas de tamaño reducido, todo ello para determinar que tarjeta de tamaño reducido se acomoda mejor para el desarrollo de este proyecto que tiene como objetivo la implementación de un dispositivo de borde la para detección de vulnerabilidades, teniendo en que las tarjetas a analizar son las más recientes disponibles en el mercado.

Tabla 2. Comparación Características Placas de Tamaño Reducido

	Hummingboard	ODROID	Banana Pi	Arduino	Raspberry Pi
Fabricante	SolidRun	Hardkernel	Sinovoip Co	Arduino	Raspberry Pi Foundation
Modelo	HummingBoard Edge	ODROID-C4	BPI-M4	Arduino YÚN	Raspberry Pi 4 Modelo B
Fecha de Lanzamiento	2016	2020	2019	2013	2019
CPU	ARM® Cortex®-A9, i.MX6Dual	Amlogic S905X3 (ARMv8-A) quad-core Cortex-A55 (2.0 GHz)	Realtek RTD1395 ARM Cortex-A53 Quad-Core 64 Bit	Atheros AR9331	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC
Frecuencia	1.2GHz	650MHz	1,8 GHz	400 MHz	1.5 GHz
Núcleos	2 núcleos	4 núcleos	4 núcleos	1 núcleo	4 núcleos
GPU	Vivante GC880/GC2000	Mali-G31 GPU MP2	Mali 470 MP4 GPU OpenGL ES 1.1/2.0	No disponible	Broadcom VideoCore VI, OpenGL ES 3.0, 1080p30 H.264/MPEG-4 AVC, 4kp60 H.265
Memoria RAM	Hasta 4GB DDR3	4GB DDR4	2 GB DDR4	64 MB DDR2	4GB LPDDR4
Pines digitales entrada/salida	36 pins GPIO Header	40 pins GPIO Header	40 pins GPIO Header	Digital I/O Pins 20 Analog I/O Pins 12 PWM Output 7	40 pins GPIO Header
USB	4 x USB 2.0	4 x USB 3.0 1 x USB 2.0	4 x USB 2.0 ports 1 x USB 2.0 Tipo C	1 x 2.0	2x USB 3.0 2x USB 2.0
Ethernet	1 x RJ-45 802.6 10/100/1000Mbps	1 x RJ45 802.6 10/100/1000 Mbps	1 x RJ45 802.6 10/100 Mbps	1 x RJ45 802.6 10/100 Mbps	1 x RJ-45 802.6 10/100/1000Mbps

Wi-Fi	Wi-Fi 802.11 a/b/g/n	Solo con adaptador USB Wi-Fi	Wi-Fi 802.11 b/g/n/AC	Wi-Fi 802.11 b/g/n	Dual-band 802.11 b/g/n/ac
Bluetooth	No Disponible	No Disponible	Bluetooth 4.2	No Disponible	Bluetooth 5.0
Almacenamiento por SD	Disponible	Disponible	Disponible	Disponible	Disponible
Sistema Operativo	Linux Windows 10 IoT Core	Ubuntu Android	Android Linux	OpenWrt-Yun	Linux Windows 10 IoT Core
Dimensiones	102mm x 69mm	85mm x 56mm	92x60mm	73 x 53 mm	85mm x 53mm
Precio	USD \$242.00	USD \$50.00	USD \$55.00	USD \$54.00	USD \$61.70

Fuente: Tomado de <https://bit.ly/3dKRdMx>, <https://bit.ly/2zCruar>, <https://bit.ly/3dYG2QP>, <https://bit.ly/2y3MC9f>

Dadas las características de las diferentes placas de tamaño reducido, se tomara como marco de trabajo la **Raspberry Pi B 4** para la implementación de este proyecto, esto debido a las características que ofrece frente a otras placas disponibles en el mercado, teniendo en cuenta en primera instancia que todas las placas presentadas son las ultimas disponibles en la actualidad, y finalmente destacar que se han tomado en cuenta algunas marcas de placas referentes en el mercado, sin embargo el respaldo que proporciona la comunidad, la alta documentación disponible, variedad de sistemas operativos, ventajas de hardware y la facilidad con la que se integrar elementos la convierte en la placa que se acomoda a la necesidad de alta demanda que se necesita para el desarrollo del proyecto.

2.2.2 Sistemas Operativos para Tarjetas de Tamaño Reducido

Al igual que muchos dispositivos encontrados en el mercado de hoy las tarjetas de tamaño reducido no es la excepción y utiliza un sistema operativo que permita gestionar de forma óptima los diferentes recursos de hardware, así mismo haga gestión adecuada de los servicios y aplicaciones, por ello, algunos de estos sistemas operativos tenidos en cuenta para efectos de este proyecto son:

Raspbian es un sistema operativo GNU/Linux gratuito basado en Debian recomendado para uso normal en una Raspberry Pi (Pi R. , Raspberry Pi, n.d.), optimizado para el hardware Raspberry Pi. Raspbian viene con más de 35,000 paquetes: software pre-compilado incluido en un formato agradable para una fácil instalación en su Raspberry Pi.

Kali Linux es una distribución de Linux basada en Debian destinada a pruebas avanzadas de penetración y auditoría de seguridad que puede utilizarse como distribución de Raspberry Pi en su edición ARM. Kali contiene varios cientos de herramientas que están orientadas a diversas tareas de seguridad de la información, como pruebas de penetración, investigación de seguridad, informática forense e ingeniería inversa (Security O. , 2019).

Kali Linux es desarrollado, financiado y mantenido por Offensive Security, una compañía líder en capacitación en seguridad de la información lo que garantiza que el

sistema esté siempre al día. Actualmente, Kali posee un alrededor de más de 600 herramientas de pruebas de penetración incluidas.

Windows 10 IoT Core es una versión de Windows 10 que está optimizada para dispositivos más pequeños con o sin pantalla que se ejecutan en dispositivos ARM y x86/x64. Microsoft lanzó al mercado en 2015 esta versión destinada para dispositivos de la Internet de las cosas (Internet of Things, IoT) como el Raspberry Pi (2 o 3). La aplicación se orienta principalmente a desarrolladores que requieran conectar los dispositivos cotidianos con Internet o crear elementos interconectados. Para tales fines, Windows 10 IoT Core se basa en la propia “Universal Windows Platform” (UWP) o API, que permite escribir aplicaciones para los dispositivos propios.

La Community Edition gratuita del Microsoft Visual Studio actúa como software de desarrollo. Aparte de eso, el sistema operativo para Raspberry Pi propietario destaca por las funciones de cifrado mediante Bitlocker y Secure Boot, que se adoptan de la versión de escritorio (Microsoft, 2015). Gracias a la modulación por ancho de pulsos (PWM, Pulse Width Modulation), con el software de sistemas también pueden controlarse, por ejemplo, electromotores.

Parrot OS Security es otra distribución para auditoría de seguridad fue liberado como una distribución especialmente pensada para computadores personales, portátiles de manera que los usuarios pudieran trabajar con ella igual que con otras distribuciones similares, Parrot Security OS cuenta con imágenes binarias optimizadas para Raspberry Pi y Cubieboard 4.

Además, también se han publicado otras dos versiones genéricas (ARMHF rootfs y ARMHF generic rootfs) de manera que puedan funcionar en otros dispositivos IoT, aunque la compatibilidad no está asegurada (Project, 2013). Puede utilizarse como servidor web de bajo consumo y construirte tu propia Smart tv, como servidor de impresión y hace una nube propia.

2.2.3 Tecnologías de Borde

2.2.3.1 Computación del Rocío

La computación del rocío soluciona principalmente problemas con los servidores o la conexión a Internet cuando no se encuentran disponibles, el usuario no puede acceder a sus datos que se encuentran ubicados en la nube. La computación del rocío o conocida también como Dew Computing permite al usuario acceder a los archivos durante momentos sin conexión a Internet (Wang, 2016); cuando se vuelve a re-establecer la conexión, los archivos y carpetas se sincronizan con el servidor en la nube.

2.2.3.2 Computación de Borde

Esta tecnología permite acercar el procesamiento a la fuente de datos y no es necesario enviarlo a una nube remota u otros sistemas centralizados para su procesamiento. Al eliminar la distancia y el tiempo que lleva enviar datos a fuentes centralizadas, podemos mejorar la velocidad y el rendimiento del transporte de datos, así como los dispositivos y aplicaciones en el borde.

Un término asociado a Edge Computing es Fog Computing. La idea detrás de ambos conceptos es la misma, acercar el procesamiento a la fuente de datos. La

diferencia entre ambas arquitecturas en el lugar en el que se ubica la capacidad de procesamiento. En Fog Computing se establece en las cercanías de la red local donde el flujo de datos es enviado hacia nodos de niebla, gateway de IoT o pequeños servidores que procesan y enrutan el tráfico hacia donde se necesite.

La computación trae consigo cierto tipo de características las cuales comprende:

- **Baja latencia:** Parte de los datos se analizan cerca de la fuente, eliminando el viaje de ida y vuelta a la nube por lo que la latencia se reduce drásticamente. Permite a las organizaciones tratar datos importantes casi en tiempo real y actuar en consecuencia, mejorando el rendimiento y la eficiencia de servicios y aplicaciones.
- **Mayor seguridad:** Cuantos menos datos almacene un sistema en un entorno en la nube, menos vulnerable será si ese entorno se ve comprometido, se produce una descentralización de los datos.
- **Menores costos:** Procesar los datos cerca de la fuente implica: eliminar lecturas erróneas, comprimirlos, darles formato, ..., reduciendo de esta forma el volumen de datos que se debe enviar a la red. Así las organizaciones podrán reducir el consumo de ancho de banda y los requisitos de almacenamiento y cómputo en infraestructuras en la nube.

2.2.3.3 Computación en la Niebla

El modelo fue introducido por Cisco System para facilitar la transferencia inalámbrica de datos a dispositivos distribuidos en el paradigma de red de Internet de las cosas (IoT).

Al igual que en la computación en la nube, la computación en la niebla o también conocida como «Fog Computing» proporciona servicios de datos, computación, almacenamiento y aplicaciones a los usuarios finales. Las características distintivas de Fog son su proximidad a los usuarios finales, su densa distribución geográfica y su apoyo a la movilidad (Systems, 2015). Los servicios se alojan en el borde de la red o incluso en dispositivos finales como decodificadores o puntos de acceso. Al hacerlo, Fog reduce la latencia del servicio y mejora la QoS, lo que resulta en una experiencia de usuario superior.

2.2.3.4 Computación en la Nube

Se basa en un modelo de computación brindado por proveedores tecnológicos por medio de internet, facilitando el acceso a hardware, software, servicios informáticos y datos ofrecidos bajo demanda (Ciberseguridad, 2017).

Antes de implementarse este modelo, las infraestructuras tecnológicas se encontraban conformadas por redes, servidores, almacenamiento, aplicaciones y servicios, que son adquiridos de forma individual, para su puesta en marcha en Datacenters, que pueden conllevar a una organización en gastos de mantenimiento, personal, recursos energéticos, etc., sin embargo todos estos elementos que conforman un infraestructura tecnológica son contratados por organizaciones y particulares, con una

inversión cada vez más accesible, garantizando la flexibilidad y facilidad de acceso desde cualquier lugar y en cualquier momento (Urueña, 2012).

Entre las características asociadas al cloud computing se encuentran las siguientes:

- Pago por uso
- Abstracción
- Agilidad en la escalabilidad
- Multusuario
- Autoservicio bajo demanda
- Acceso sin restricciones
- Seguridad

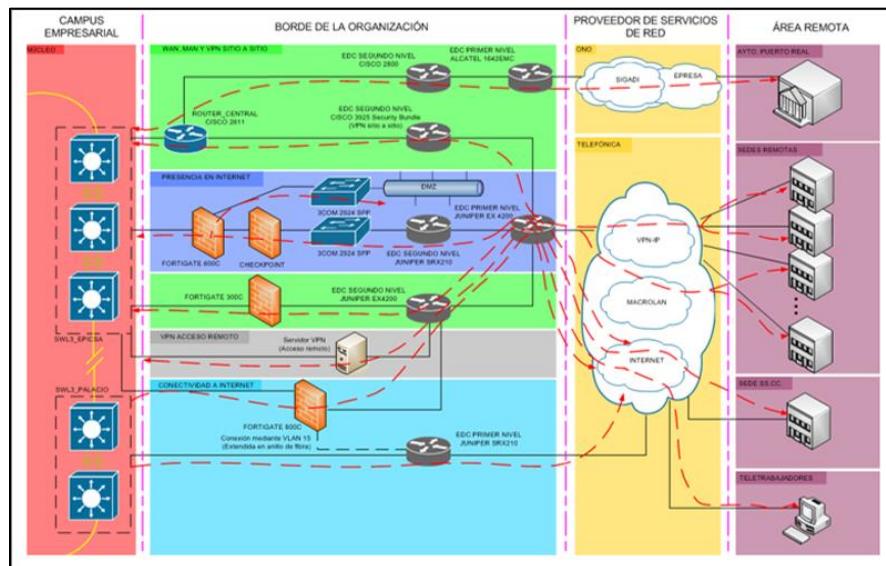
2.2.3.5 Red de Borde

El núcleo de una revolución digital se encuentra en el borde de la red, igualmente conocido como network edge, que es la capa de una infraestructura de red en la cual se realiza la conexión de los dispositivos hacia la red empresarial o a internet.

Se tienen en cuenta laptops, smartphones, tablets o dispositivos de diferentes características, el borde de red es el punto de acceso a la información del negocio en una organización y, por este motivo, una localización crítica para optimizar la experiencia de los usuarios. Independientemente de si es la una red LAN, inalámbrica o inclusive el borde de una red en sucursal o WAN, el borde es hoy el catalizador del éxito en los negocios más que nunca (Mendez, 2017).

La red de borde en una organización reúne la conectividad de varios dispositivos externos al campus de la organización y enruta el tráfico hacia la capa de núcleo de la infraestructura de red interna (Carretero Aguilar, et al., 2019). Los dispositivos pertenecientes al área de borde de la organización ofrecen la capacidad de seguridad que permiten de esta forma estabilidad correcta de los recursos de la organización cuando se producen conexiones con redes públicas y/o Internet. La topología lógica del área de borde de la organización, junto con el extremo del ISP y el área remota se puede evidenciar en la Ilustración 10.

Ilustración 10. Red de Borde



Fuente: Tomado de <https://bit.ly/3c1LvFX>

2.2.3.6 Dispositivos de Borde

La seguridad en una infraestructura de la red compuesta por routers, switches, servidores, estaciones de trabajo y otros dispositivos, es crítica en borde de una red, esto pues se encuentra expuesta en internet.

El modelo actual se fundamenta en un router de borde quien es el último entre la red interna e Internet. Todo el tráfico a Internet de una organización pasa por este router de borde; por consiguiente, habitualmente funciona como la primera y última línea de defensa de una red. A través del filtrado inicial y final, el router de borde ayuda a asegurar el perímetro de una red protegida (Systems, Cisco CCNA Security 1.0, 2012). Del mismo modo es responsable de implementar las acciones de seguridad que están basadas en las políticas de seguridad de la organización.

La implementación de este router de borde se centra en tres enfoques:

- Enfoque de un solo router
- Enfoque de defensa profunda
- Enfoque DMZ

En el **enfoque de un solo router**, este enlaza la red protegida, o LAN interna a Internet. Cada una de las políticas de seguridad se encuentran configuradas en este dispositivo. Por lo general se utiliza este diseño en implementaciones de sitios pequeños como sucursales y SOHO (Ariganello, 2014). En las redes más pequeñas, las funciones de seguridad requeridas pueden ser soportadas por ISR sin comprometer el rendimiento del router.

El **enfoque de defensa profunda** es mucho más seguro que el de un solo router. En este enfoque, el router de borde se maneja como la primera línea de defensa y se lo conoce como screening router. Envía al firewall todas las conexiones dirigidas a la LAN interna. El segundo método de defensa es el firewall. El firewall básicamente retoma donde dejó el router y realiza filtrado adicional. Suministra control de acceso adicional esto debido a que monitorea el estado de las conexiones, actuando como un dispositivo

de control. El router de borde tiene un conglomerado de reglas que especifican qué tráfico debe permitir acceso y qué tráfico se debe denegar.

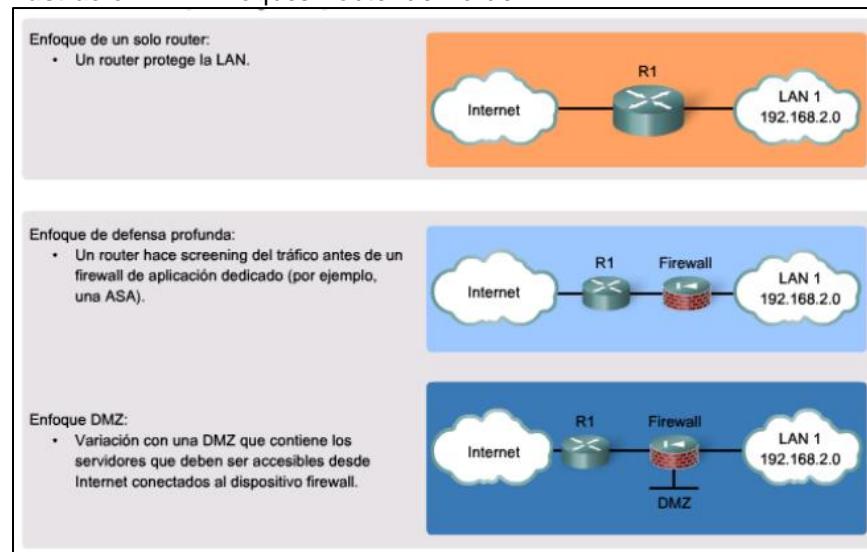
De forma predeterminada, el firewall deniega la iniciación de conexiones desde las redes externas (conexiones no confiables) para la red interna (conexiones confiables). Aunque, permite a los usuarios internos conectarse a las redes no confiables y permite que las respuestas vuelvan a través del firewall (Systems, Cisco CCNA Security 1.0, 2012). De igual forma puede hacer autenticación de usuarios (proxy de autenticación) para que los usuarios tengan que estar autenticados para ganar acceso a los recursos de red.

El **enfoque DMZ** se centra en brindar un espacio intermedio llamado zona desmilitarizada (demilitarized zone - DMZ). Esta zona puede ser utilizada para los servidores que tienen que ser accesibles desde Internet o alguna otra red externa. La DMZ puede ser impuesta entre dos routers, con un router interno conectado a la red protegida y un router externo conectado a la red no protegida, o ser simplemente un puerto adicional de un solo router (Ariganello, 2014).

El firewall, situado entre las redes protegida y no protegida, se instala para permitir las conexiones requeridas (por ejemplo, HTTP, conexiones a correo, FTP) de las redes externas (conexiones no confiables) a los servidores públicos en la DMZ. EL firewall sirve como defensa primaria para todos los dispositivos en la DMZ. En el enfoque DMZ, el router provee protección filtrando algún tráfico, pero deja la mayoría de la protección a cargo del firewall.

Todos los enfoques mencionados con anterioridad pueden ser evidenciados en la Ilustración 11 que a continuación se muestra.

Ilustración 11. Enfoques Router de Borde



Fuente: Tomado de CCNA Security Capítulo 2

2.2.4 Industria 4.0

El término industria 4.0 se utiliza de manera generalizada en Europa, si bien se acuñó en Alemania en el año 2011 (del Val Román, 2016), la industria 4.0 se define como un nuevo modelo de organización industrial que se orienta a la autoorganización y la autogestión de sistemas de producción con un enfoque altamente de automatización

(Didiana Velásquez, 2019), permitiendo que dispositivos se comuniquen entre ellos y aprendan de forma autónoma, todo ello mientras se lleva a cabo el ciclo de vida un producto.

Es común además referirse a este concepto con términos tales como "Fábrica Inteligente" o "Internet industrial". En síntesis, se trata de la aplicación a la industria del modelo "Internet de las cosas" (IoT). Cada uno de estos términos tienen en común el reconocimiento de avances en la fabricación de productos gracias a la transformación digital, las cuales permiten la vinculación del mundo físico (dispositivos, materiales, productos, maquinaria e instalaciones) con el digital (sistemas) para convertirlo en una nueva "revolución industrial" (Blanco, Fontrodona, & Poveda), en donde las barreras entre las personas y las máquinas se difuminan.

Entre los siglos XVIII y XIX se desarrolló la primera Revolución Industrial, en donde los procesos de producción se mecanizaron, transformando la economía agraria y artesanal en otra liderada por la industria. La siguiente transitividad de la industria tuvo lugar en el siglo XX, que trajo la producción en serie, con la aparición de fábricas y líneas de montaje que permitieron fabricar productos para el gran consumo. Para finales del siglo XX se produce la tercera revolución industrial (del Val Román, 2016). La evolución de la electrónica y la informática aplicada a la industria permitió automatizar de actividades repetitivas a líneas de producción por medio de las máquinas.

Como se ha podido evidenciar, en las tres revoluciones industriales que se han presentado en los últimos 250 años se han implementado aquellas tendencias que en su momento marcaron la diferencia y fueron acomodadas a la industria para facilitar la realización de productos (Castresana Sáenz, 2016).

Todo ello ha promovido descubrimientos y avances tecnológicos a nivel global que han marcado cambios de ritmo agigantados diversos ámbitos y cuya aplicación cada vez mayor desencadenó incrementos de productividad, ahorros en tiempos de fabricación, mejoras de la eficiencia y aumento de beneficios. En la Tabla 3 se especifica la arquitectura tecnológica y de soporte que forman parte de la industria 4.0 que se vive en la actualidad.

Tabla 3. Perspectiva Bajo la Industria 4.0

Industria 4.0			
Fábricas Inteligentes	Ciudades Inteligentes	Productos Inteligentes	Servicios Inteligentes
Artefactos Tecnológicos Integrados			
Sensores, Microchips, Sistemas Autónomos, Sistemas Ciberfísicos, Maquinas Autónomas			
Características			
Inteligencia, Flexibilidad, Conectividad, Seguridad, Confiabilidad, Trazabilidad, Movilidad, Colaboración, Sociabilidad, Sustentabilidad			
Principios de Diseño			
Integración, Interoperabilidad, Virtualización, Descentralización, Capacidad de Tiempo Real, Orientación al Servicio, Modularidad			
Arquitectura de Soporte			
Internet de las Cosas (IoT), Identificación por Radiofrecuencia (RFID), Redes Industriales,			

Computo de Alto Desempeño (HPC), Computo Móvil, La Nube y el Internet de los Servicios (Infraestructura (IaaS), Plataforma Tecnológicas (PaaS) y Software (SaaS) Como Servicio), Big Data y Analítica Avanzada

Beneficios

Producción orientada a la demanda, uso más eficiente de los recursos, productividad, reducción de costos, ciclo de desarrollo de producción más cortos, mayor competitividad, optimización de los procesos, autonomía en la toma de decisiones y cadenas de suministros más integradas

Fuente: Tomado de <https://bit.ly/3e1tBDQ>

De acuerdo con Warren Bennis (Castresana Sáenz, 2016), “*la fábrica del futuro tendrá dos empleados: un humano y un perro. La labor del humano será dar de comer al perro y la del perro, evitar que el humano toque los sistemas automatizados*”, esta reflexión toma sentido en un mundo actual, en donde casi todas las organizaciones empiezan a adoptar la industria 4.0 y se evidencia todas y cada una de las tecnologías que la componen, donde estas abarcan entornos digitales hasta aquellos físicos que se consideraban que solo eran posible realizar por interacción humana, por cual muchos sectores ya están implementando la automatización como se muestra en la Ilustración 12, en donde se puede apreciar que varios sectores han acogido en casi un 80% esta nueva industria.

Ilustración 12. Industria 4.0 por Sectores



Fuente: Tomado de <https://bit.ly/2ZeYSPd>

Ahora bien, la industria 4.0 no solo afecta modelos de negocio en empresas, sino es inmersa en la cotidianidad de los seres humanos, todo ello mediante la introducción de nuevas tecnologías y procesos, entre las que se encuentra la ciberseguridad, realidad aumentada, computación en la nube, robots autónomos, internet de las cosas, análisis big data, simulación y por último la fabricación aditiva.

En ese mismo sentido, la cuarta revolución industrial es una revolución hacia la digitalización. El cambio social de una sociedad industrializada a una sociedad post industrializada, establecido en el conocimiento, orientada al servicio, y situado en la información y datos que se puedan generar, puede ser denominado como una revolución digital (Didiana Velásquez, 2019).

Por ello, estamos al comienzo de la cuarta revolución industrial que se centra en una revolución digital con Internet mucho más móvil, sensores más pequeños y potentes que cada vez se hacen más baratos y asequibles, e inteligencia artificial y aprendizaje automático. Este cambio significa aprovechar al máximo la conectividad y utilizar los datos más allá de los límites (Voices, n.d.).

2.2.5 Internet of Things (IoT)

Tras el avance sustancial del protocolo TCP/IP desarrollado por la década de los 70's aplicada a dispositivos de cómputo en 1990, John Romkey en el evento Interop en EEUU, creó el primer objeto conectado a Internet; una tostadora que se podía encender o apagar a través de internet (Cendon, 2017).

La conectividad fue a través del ya mencionado protocolo TCP/IP y el control se realizó mediante el protocolo SNMP (Simple Network Management Protocol), protocolo de gestión de red. Acorde a Cisco Systems y su Grupo de Soluciones Empresariales Basadas en Internet (IBSG, Internet Business Solutions Group) (Systems, CISCO, 2011), IoT nació entre 2008 y 2009, momento en el cual se conectaron a Internet más cosas que personas habitando en el mundo.

El término Internet de las Cosas fue empleado por primera vez en 1999 por el británico Kevin Ashton para describir un sistema en el cual los objetos del mundo físico se podían conectar a Internet por medio de sensores (Karen Rose, 2015).

Permitiendo así la adquisición de datos y la entrega de órdenes a dispositivos que interactúan o forman parte del mundo real (Andrés, 2018), en el cual reconocen eventos y cambios, y pueden reaccionar de forma autónoma y acertada.

De igual forma, a lo largo de los años se han acuñado otros términos, tales como el que propone Omar Said (Said & Mehedi Masud, 2013) en el cual indica que es una tecnología que implica el proceso de conectar máquinas, equipos, software y cosas, permitiendo así la comunicación entre ellos, sin necesidad de intervención humana.

En la actualidad, IoT puede ser visto como una combinación de sensores y actuadores que son capaces de proporcionar y recibir información digitalizada y colocarla en redes bidireccionales capaces de transmitir todos los datos para ser utilizados por una gran cantidad de diferentes servicios y usuarios finales (Karen Rose, 2015).

Gran variedad de sensores se pueden unir a un objeto o dispositivo para medir una amplia gama de variables físicas o fenómenos y luego transmitir todos los datos a la nube. La detección puede ser entendida como un modelo de servicio (Salazar & Silvestre, 2016). Estos dispositivos constan de una clasificación la cual se puede observar en la Tabla 4.

Tabla 4. Clasificación de Sensores IoT

Clasificación de Sensores	
Proveedores de datos del sensor	Las entidades empresariales que implementan y administran por sí mismos sensores.
Organizaciones	Público o Privado. Infraestructuras públicas. Las organizaciones comerciales.

	Corporaciones privadas: los proveedores de tecnología y servicios.
Personal y Hogares	Los teléfonos móviles, relojes inteligentes, giroscopios, cámaras, GPS, acelerómetros micrófonos, ordenadores portátiles, alimentos y artículos para el hogar, tales como televisores, cámaras, congeladores, hornos de microondas, lavadoras, electrodomésticos inteligentes, etc.

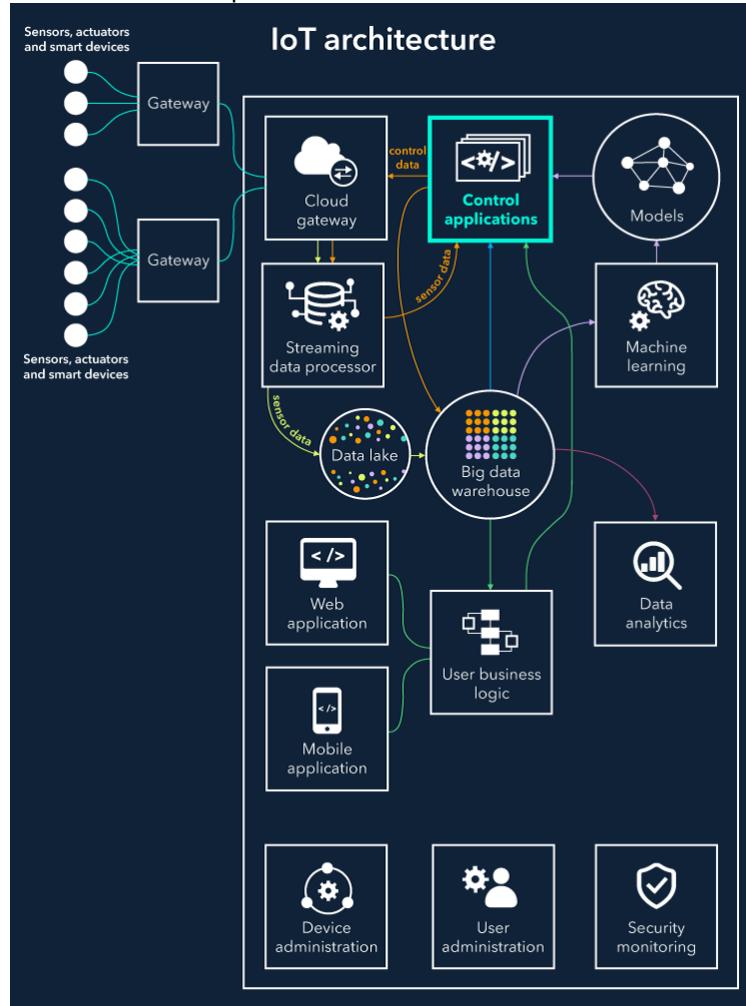
Fuente: Tomado de <https://bit.ly/3bb79qG>

2.2.5.1 Arquitectura del IoT

Los diversos dispositivos conectados entre sí que incluyen sensores, se fundamentan con base en la capa física de baja velocidad que se denomina Internet 0, en esta cada uno de los dispositivos cuentan con la capacidad de conectarse a internet y poder así transmitir e intercambiar información a baja velocidad (Baquero Rey & Hernández Bejarano, 2018).

Así, la transmisión de la información es gracias a que se establecen lineamientos (ver Ilustración 13) que permiten procesar los eventos que se generen, en donde toman cambios del estado del dispositivo (por ejemplo, encender o apagar la luz, abrir o cerrar una puerta, aumentar o disminuir velocidad de rotación del motor y más) y tendrán la capacidad de forma paulatina de generar sus propias alertas que podrán ser enviadas al usuario final para que tome las medidas necesarias para el evento.

Ilustración 13. Arquitectura IoT



Fuente: Tomado de <https://bit.ly/36jc2fJ>

La arquitectura IoT no solo está constituida por las tantas “cosas” conectadas entre sí, también hacen parte algunos elementos que hacen que la experiencia del usuario sea mejor y permita una comunicación entre todos los dispositivos, entre los elementos se encuentran:

Canal: Es el medio de transmisión por donde se envían los eventos generados por cada uno de los dispositivos IoT, aunque se debe tener presente que no siempre se tiene un canal alojado en Internet, también se puede estar hablando de una canal de transmisión lógico, en donde se haga la entrega de datos de forma bidireccional entre la nube del fabricante y el consumidor (Moreno Saiz, 2915), todo ello mediante la conexión de nivel de transporte haciendo uso de protocolos TCP o UDP.

Local Gateway: Los datos van de las “cosas” a la nube y viceversa a través de las puertas de enlace o también llamada “gateways”. Este elemento brinda la conectividad entre los dispositivos conectados de la LAN (Local Area Network) y la parte del cloud de IoT, permitiendo así el procesamiento y el filtrado de datos antes de moverse a la nube haciendo “limpieza” de los datos para el almacenamiento y procesamiento adecuado.

Cloud Gateway: Proporciona la transmisión segura y facilidad de comprensión de los datos generados por los sensores y demás dispositivos de IoT entre el gateway y los

servidores de IoT en la nube. Además, asegura la compatibilidad con varios protocolos y se comunica con los gateway locales.

Streaming Data Processor: Asegura la transmisión correcta de los datos de entrada al denominado “data lake” y “control applications”.

Data Lake: Es utilizado para almacenar los datos generados por los dispositivos. Cuando es necesario obtener información significativa, se extraen de este elemento y se cargan/almacenan a la “big data”.

Big Data: Los datos filtrados y procesados por los “local gateways” pasan al “data lake” y finalmente se carga datos “limpios”, estructurados y coincidentes a la Big Data. Así mismo, la correlación y patrones encontrados manualmente pueden contribuir a la creación de algoritmos para los “control applications”.

Machine Learning y Modelos ML: Gracias al Machine Learning existe la posibilidad de crear modelos más precisos y más eficientes para los “control applications”. Los modelos se actualizan con regularidad (por ejemplo, una vez a la semana o una vez al mes) en función de los datos históricos aglomerados en la Big Data.

Control Applications: Envían comandos y alertas de forma autónoma a los actuadores, como por ejemplo:

- Una casa inteligente puede recibir un comando automático para abrir o cerrar las ventanas, todo de acuerdo a los pronósticos tomados del servicio meteorológico.
- Cuando los sensores detectan que el suelo está seco, los sistemas de riego obtienen un comando automático para regar las plantas.
- Los sensores pueden monitorear el estado de un vehículo o un equipo industrial y en determinado caso de detectar posible avería a futuro, un sistema IoT genera y envía alertas a los ingenieros de campo.

Las “Control Applications” pueden estar basadas en reglas o Machine Learning. En el primer caso, las aplicaciones de control funcionan tomando las reglas previamente configuradas por los especialistas. En el segundo caso, las aplicaciones de control están utilizando modelos que se actualizan con frecuencia como ya se comentaba con anterioridad. En todo siempre es conveniente por una opción los usuarios influyan en el comportamiento de dichas aplicaciones, por ejemplo, en caso de emergencias o que el dispositivo IoT genere comportamientos erráticos.

User Applications: Permiten la conexión entre el componente de software de un sistema IoT y su usuario final (Grizhnevich, 2018), así mismo brinda las opciones de monitorear el estado de los dispositivos, enviar comando de control y establecer configuraciones de comportamiento, todo ello gracias a los diferentes asistentes virtuales que se integran y ofrecen en la actualidad disponibles en aplicaciones móviles o la web.

2.2.5.2 Protocolos Asociados a IoT

La Real Academia Española (RAE) define protocolo como un conjunto de reglas que se establecen en el proceso de comunicación entre dos sistemas (Protocolo, n.d.), siendo este concepto utilizado desde la creación de Internet para estandarizar aquellos lineamientos que fueron la base fundamental para que nuevas tecnologías se integraran

sin dificultad y pudieran dar extensión a esta gran red, por ejemplo, TCP/IP, fue uno de ello. Pese a ello con la llegada del internet de las cosas (IoT) se ha convertido en un gran desafío la interoperabilidad de estos dispositivos (Semle, n.d.).

Existen varios protocolos para complementar esto; algunos son privados y otros que son estándares abiertos. Todas las empresas productoras de dispositivos IoT están compitiendo para generar y convertir uno sus protocolos en único de IoT, pero es claro que eso nunca será una realidad. Estos protocolos coexistirán, en donde cada uno posee sus propias fortalezas y debilidades. A continuación, en la Tabla 5 se muestran algunos protocolos relevantes.

Tabla 5. Protocolos Asociados a IoT

Protocolo	Descripción
Zigbee	Basado en el estándar 802.15.4 de IEEE usando la banda 2.4GHz y una red de malla real de autocuración; Zigbee se convierte en un conjunto de protocolos de alto nivel de comunicación inalámbrica (Alliance, n.d.).
Z-Wave	Permite que productos para el hogar inteligente como cerraduras, luces y termostatos se comuniquen entre sí. Esto crea la columna vertebral de un hogar inteligente y permite el uso de teléfonos inteligentes o tablets (Z-Wave, n.d.).
UPnP (Universal Plug and Play)	Es un conjunto de protocolos [UDP, HTTP] propuesta por Microsoft y promulgada por el UPnP Forum que permite que varios dispositivos de red se configuren por sí mismos (Alarcón, González, & González, 2017).
AllJoyn	Es un protocolo de código abierto que fue lanzado por The AllSeen Alliance, organizado por Haier, LG, Microsoft, Panasonic, Qualcomm, Sharp, Silicon Image, Technicolor y TP-Link. Facilita la comunicación entre dispositivos y aplicaciones, para todo tipo de protocolos de la capa de transporte [TCP, UDP].
HomePlug y HomeGrid	Son protocolos cuya comunicación se realiza a través de la red eléctrica. Dependiendo del producto adquirido, el tipo de cifrado es diferente, incluso algunos dispositivos transmiten la información sin cifrar.
MFi (Made For iPhone/iPod/iPad)	Es un protocolo de comunicaciones exclusivo de Apple. Los dispositivos y elementos de conexión de Apple incorporan un chip mediante el cual verifican que tanto los dispositivos propietarios de su marca, como los cables de conexión son originales.
OCF (Open Connectivity Foundation)	Es un proyecto de código abierto que ofrece interconectividad con la filosofía just-works. Es un protocolo que permite que los dispositivos se comuniquen independientemente del factor de forma, sistema operativo, proveedor de servicios, tecnología de transporte o ecosistema, incluyendo IoT.
Thread (network protocol)	Fue creado por el conjunto de empresas denominado Thread Group. Es una tecnología basada en las comunicaciones por red mediante IPv6 que utiliza cifrado

	AES. Por ello y por la flexibilidad que ofrece, es un protocolo seguro.
MQTT (MQ Telemetry Transport)	Es un protocolo PubSub de Message Service que actúa sobre TCP, creado en 1999. Se caracteriza por ser ligero, sencillo de desplegar y consumo bajo de ancho de banda. Es conveniente para dispositivos de baja capacidad como los que habitualmente se hace uso en IoT.
AMQP (Advanced Message Queuing Protocol)	Es un protocolo PubSub de Message Queue. AMQP está diseñado para garantizar la confiabilidad e interoperabilidad. Está pensado para aplicaciones corporativas, con rendimiento superior y redes de baja latencia. No resulta tan adecuado para aplicaciones de IoT con dispositivos de bajos recursos.
WAMP (Web Application Messaging Protocol)	Es un protocolo abierto que se ejecuta sobre WebSockets, y provee tanto aplicaciones de PubSub como rRPC.

Fuente: Tomado de <https://bit.ly/2WzYCr7> y <https://bit.ly/2yFyAuL>

2.2.5.3 Modelos de Implementación IoT

Las opciones de implementación para IoT difieren ampliamente según su aplicación, industria y uso definido. Sin embargo, generalmente se pueden clasificar las implementaciones de IoT en una de tres formas: **device-to-device**, **device-to-cloud** o **device-to-gateway**.

El modelo **device-to-device** representa dispositivos que descubren, conectan y comunican directamente utilizando las redes disponibles localmente (Hosmer C. , 2018). La comunicación puede ser a través de las redes tradicionales TCP (Protocolo de control de transacciones) / UDP (Protocolo de datagramas de usuario) /IP (Protocolo de Internet); pero, en muchos casos, se comunican a través de redes inalámbricas o de baja potencia como Bluetooth, Z-Wave, ZigBee y Universal Plug and Play (uPnP).

Los dispositivos IoT que utilizan el método **device-to-cloud** se conectan directamente a un servicio en la nube basado en Internet para intercambiar datos y controlar mensajes. Este método generalmente utiliza protocolos tradicionales como TCP, UDP, HTTP, HTTPS y TLS (Seguridad de la capa de transporte) para intercambios basados en la seguridad (Hosmer C. , 2018).

Por último, haciendo uso del modelo **device-to-gateway**, los sensores descubren y se comunican con otros sensores y coordinan información a través de puertas de enlace (Software O. A., 2020). La puerta de enlace o gateway, a su vez, comunica información con otras redes de sensores y típicamente con la nube.

2.2.5.4 Aplicaciones del IoT

El IoT se ha convertido en el eje central de la sociedad actual, donde ha quedado inmersa en la cotidianidad de las personas, brindándoles ayudas y facilidad a la hora de realizar cierto tipo de actividades, siendo tan influyente que al igual que ha afectado el estilo de vida de la sociedad se encuentra aplicada en diversos sectores económicos, algunos de ellos son:

- Agricultura/ganadería
- Salud
- Medio Ambiente
- Vehicular

En el sector de **agricultura/ganadería** se observa un panorama altamente productivo (Duran Caastillo, 2019), en donde se habla de la implementación de smart tractors, siendo estos tractores inteligentes que sustituyen la interacción física del humano por un sistema autónomo basado en cámaras, radares, GPS y sensores que detectan obstáculos, así mismo se habla de la utilización de drones, ganadería conectada, control de plaga inteligente y sistema de riegos inteligente, todo ello conectado entre sí y permitiendo a través de un monitoreo en línea a los agricultores por medio de smartphones o tablets conocer la temperatura, humedad, estado y factores físicos que le permitan tomar las mejores decisiones para sus cultivos o ganado.

El internet de las cosas aplicada en el campo de la **salud** toma un nuevo término denominado “Internet of Medical Things (IoMT)” donde se estima que alcanzará para el año 2021 la cifra de 136.8 mil millones de dólares, siendo actualmente 3.7 millones la cifra de dispositivos médicos en uso conectados y monitorizando datos médicos.

Muchos de estos dispositivos serán biosensores que permitirán tener un monitoreo de un paciente, en donde la capacidad de conectividad entre dispositivos será fundamental gracias a la capacidad de proporcionar un grado de movilidad (Salud, 2018), así mismo se convierte en una herramienta fundamental esto pues hay enfermedades que son asintomáticas y muchas otras presentan un diagnóstico tardío (Cera Cárdenas, Martínez Otero, Rojas Blandón, Villaveces Santander, & Sanmartín Mendoza, 2015), estos sensores permitirán detectar un diagnóstico temprano, mitigar síntomas y prevención de enfermedades.

La degradación actual que presenta el **medio ambiente** debido al desgaste de recursos naturales es evidente. En Estados Unidos, los edificios consumen el 70% de toda la electricidad, de la cual un 50% se malgasta. Igualmente, un 50% del agua que consumen también es derrochada, convirtiendo en un problema a largo plazo, en donde diferentes sensores conectados entre sí pueden ser la solución, estos se encuentran dotados para recoger información tomando como base factores ambientales como la calidad del aire, suelo, agua y polución sonora (Alcaraz, 2014), convirtiéndose en lo que se conoce como edificios “smart grid” (Bankinter, 2011), un conglomerado de sensores inteligentes que permite optimizar el consumo de los recursos naturales y se evidencia una reducción en aspectos económicos.

De acuerdo con la empresa multinacional Willis Tower Watson (Watson) encargada de la gestión de riesgos, muestra un futuro en donde los **vehículos** se asentaran en el pilar del internet de las cosas, donde el manejo de datos en tiempo real proporcionado por los sensores instalados en cada vehículo, permitirá tener un control de las rutas, permitiendo anticipar variables en el tráfico mejorando tiempos de entregas y llegadas a destino, además factores humanos serán reconocidos, como la detección de cansancio de un conductor.

Además, la aplicación en este campo es muy amplia en el cual se pueden detectar averías de forma anticipada, respuesta rápida y oportuna en caso de accidentes y presenta la modalidad de sincronización con smartphones dando la posibilidad de configurar de forma personalizada aspectos como temperatura ambiente, música de fondo

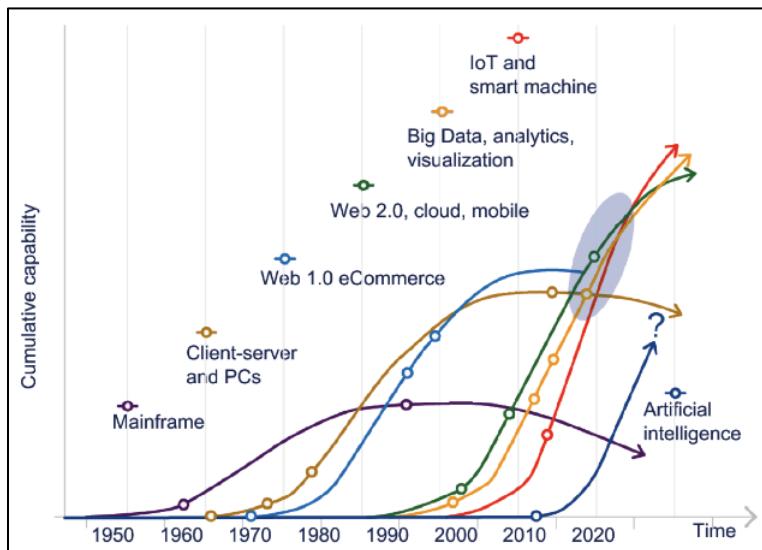
(Baquero Rey & Hernández Bejarano, 2018), hasta la posibilidad de conducción y aparcado de forma autónoma.

2.2.5.5 Retos del IoT

Diversas empresas y organizaciones dedicadas a la investigación han publicado una amplia gama de proyecciones sobre el potencial impacto que tendrá la IoT sobre Internet y sobre la economía en los próximos cinco a diez años. Como ejemplo, Morgan Stanley anticipa que para el año 2020 habrá 75 mil millones de dispositivos conectados en la red. Considerando un período de tiempo más largo, Huawei sube la apuesta y anticipa que en 2025 habrá 100 mil millones de conexiones a la IoT.

El McKinsey Global Institute sugiere que el impacto financiero de la IoT sobre la economía global puede llegar a ser de \$3.9 a \$11.1 mil millones en 2025. Aunque la variabilidad de las predicciones las vuelve cuestionables, en conjunto permiten entrever una influencia y un crecimiento significativos (Karen Rose, 2015) como se muestra en la Ilustración 14 proporcionada por World Economic.

Ilustración 14. Tendencia IoT Año 2020



Fuente: Tomado de <https://bit.ly/2WGHoOY>

De acuerdo lo anterior, el crecimiento exponencial que abarcara IoT en el ambiente doméstico y empresarial será impresionante, en donde se encontraran diversos factores de riesgos que puedan afectar en gran medida la privacidad de información de los consumidores, posibles intrusiones y explotaciones de vulnerabilidades en estos objetos conectados a internet que puedan poner en peligro la integridad de una persona, por ejemplo, un automóvil conectado a internet puede ser controlado por un cibercriminal que puede de forma mal intencionada hacer funcionar de forma errónea la dirección del volante o en casos fatales no funcionar los frenos del automóvil causando accidentes fatales (Winder, 2020).

De igual forma IoT se ha adentrado a campos que hasta el momento se habían pensado inimaginables, en los últimos años ha surgido el termino IoNT siendo siglas de la definición “Internet de las Nano Cosas” (Maksimović, 2017), siendo así que ahora se habla de nanotecnología aplicada en IoT, dejando así su amplia utilización en medicina y atención médica tiene el potencial de brindar los mayores beneficios a la sociedad.

Aunque serán tecnologías emergentes que brindaran ayudas a la humanidad se debe tener en presente que se han convertido en un gran aliado de algunos grupos que pretender intervenir en un mundo de vigilancia de masas, siendo herramienta adecuada para estos escenarios, por ello, radica la necesidad de fortificar la seguridad en el internet de las cosas garantizando así la privacidad, que ya se convierte en computación ubicua en donde la tecnología no se percibe como objeto diferenciador, y se logra el acceso a la información en cualquier momento y a través de diversos dispositivos conectados (Sánchez Martelo, 2015), en donde se encuentra inmersa la tecnología y el ser humano, dando paso a la omnipresencia de los sistemas de información.

2.2.6 Seguridad Informática

Antes de adentrarse en detalle en términos informáticos es importante entender que la seguridad de acuerdo a la Real Academia Española (RAE) proviene del latín *securitas* haciendo énfasis a la característica de seguro, permitiendo un estado de bienestar y ausencia de riesgo debido a la confianza existente en alguien o algo, siendo este un término interdisciplinario que permite evaluar y hacer frente a los riesgos que son expuestos una persona, un animal, el ambiente o un bien (3Ciencias, 2018).

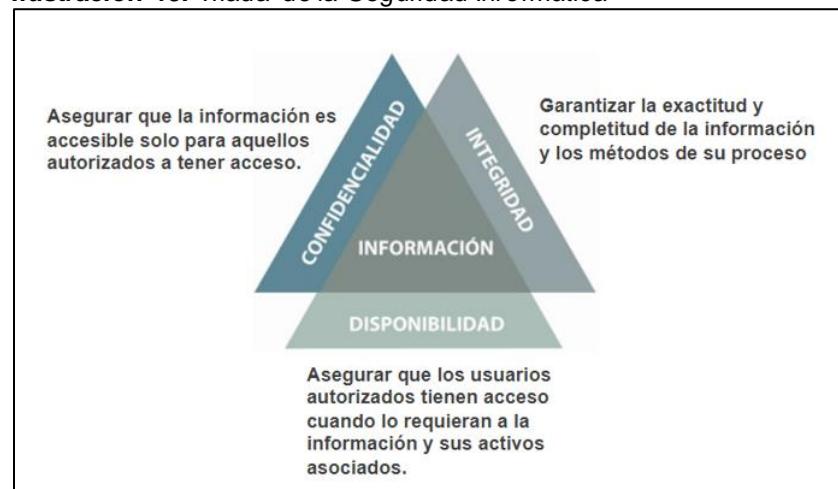
La seguridad busca la gestión de riesgos, permitiendo así evitarlos o prevenirlos realizando acciones para contrarrestar situaciones de la mejor forma.

Por otro lado, la seguridad informática hace referencia a cualquier acción no autorizadas sobre un sistema informático o red de computadoras, dando así la garantía de protección en activos.

En líneas generales, comprende el conjunto de medidas preventivas, de detección y corrección destinadas a proteger los recursos informáticos de una organización (Sain, 2018). De los diferentes factores que inciden en la seguridad informática, se podría definir como los más importantes la confidencialidad, la autenticidad y la integridad.

En la Ilustración 15 se muestra la triada denominada CIA (Confidentiality, Integrity, Availability) como se explicaba con anterioridad, en donde cada uno de los aspectos cumple una función a la hora salvaguardar la información.

Ilustración 15. Triada de la Seguridad Informática



Fuente: Tomado de <https://bit.ly/3ekJaqY>

La protección se lleva a cabo en estos medios informáticos a través de estándares, normativas, protocolos de comunicación seguros, estándares, reglas, herramientas y leyes locales garantizando así minimizar el riesgo que impacten económicamente o de forma reputacional una organización, en donde se quieren proteger activos o los recursos que forman parte del sistema en donde se encuentran los siguientes:

- **Hardware:** elementos físicos de computación, tales como procesadores, electrónica y cableado de red, medios de almacenamiento (cabinas, discos, cintas, DVDs, ...).
- **Software:** elementos lógicos informáticos, comprende el conjunto de programas que se ejecutan sobre el hardware, tal como el propio sistema operativo y las aplicaciones instaladas en él.
- **Datos:** información administrada por el hardware y el software, hace parte la información lógica que procesa el software haciendo uso del hardware. En general serán informaciones estructuradas en bases de datos o paquetes de información que viajan por la red (INTEF, n.d.).
- **Otros:** fungibles, personas, infraestructuras...

De acuerdo con el Departamento de Seguridad Nacional de EEUU, nuestra vida diaria, vitalidad económica y seguridad nacional dependen de un ciberespacio estable, seguro y resistente. El ciberespacio y su infraestructura subyacente son vulnerables a una amplia gama de riesgos derivados de amenazas y peligros físicos y cibernéticos (Security D. O.).

A lo anterior, la seguridad informática cumple un papel fundamental implicando así estrategias que vayan un paso a la acción de un atacante, garantizando así la seguridad en infraestructuras tecnológicas, los usuarios y por supuesto la información guardada por cada uno de estos.

Un estudio realizado por Panda Security, empresa dedicada a la creación de soluciones de seguridad informática presenta un panorama en donde 43% de los ciberataques afectan a pequeños negocios (Security P., Panda Security, n.d.), siendo un punto de partida para tomar conciencia de la importancia a la hora de implementar la seguridad informática y que contengan estrategias de controles que contemplen desde los entornos más pequeñas hasta los más grandes, teniendo presente que es un problema global.

2.2.7 Vulnerabilidades

Una vulnerabilidad (en términos de informática) es una debilidad o fallo en un sistema de información teniendo como resultado riesgos a la seguridad de la información pudiendo permitir que un atacante pueda comprometer la integridad, disponibilidad o confidencialidad de esta, así pues, es necesario encontrarlas y eliminarlas lo antes posible (INCIBE, 2017).

Cada vez las vulnerabilidades informáticas reportadas y encontradas por profesionales dedicados a la seguridad informática son muchas más abundantes y en algunos casos críticas en diferentes tecnologías disponibles en el mercado. Cifras entregadas en el año 2018 por la compañía de seguridad informática ESET observables la Ilustración 16 se reportaron 16029 vulnerabilidades, significando así un incremento del 9% respecto al año 2017. La cifra deriva un promedio 46 vulnerabilidades reportadas por día durante el 2018.

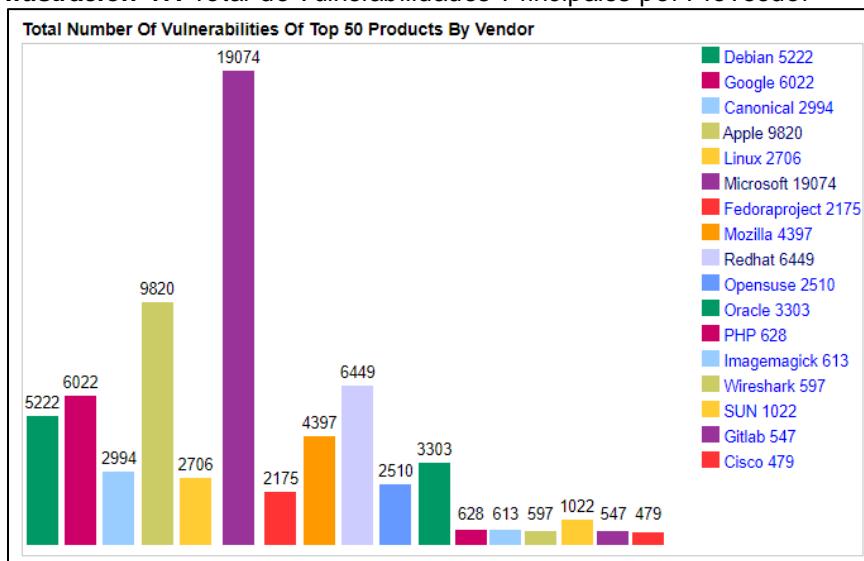
Ilustración 16. Histórico de Vulnerabilidades Año 2018

Fuente: Tomado de <https://bit.ly/3ep628C>

CVE Details es una web que proporciona una interfaz web fácil de usar para la búsqueda de datos relacionados con vulnerabilidades. En esta web es posible buscar proveedores, productos y versiones y ver entradas de CVE, vulnerabilidades relacionadas con ellos.

De acuerdo a las estadísticas entregadas por CVE Details en su reporte denominado “los 50 productos principales por número total de vulnerabilidades” que se puede apreciar en la Ilustración 17, en donde se evidencia una alta tasa de vulnerabilidades descubiertas para Microsoft con un total de 19074 vulnerabilidades encontradas desde el año 1999 hasta la fecha, predominando la vulnerabilidad de ejecución de código.

En un segundo lugar se encuentra Apple con un total de 9820 vulnerabilidades, que al igual que Microsoft predomina la vulnerabilidad de ejecución de código. Finalmente, como tercer lugar Redhat con 6449, contando con un alto volumen de reportes de vulnerabilidades relacionados a denegación de servicio.

Ilustración 17. Total de Vulnerabilidades Principales por Proveedor

Fuente: Tomado de <https://bit.ly/3osmPzl>

En la década de los 80's las vulnerabilidades de seguridad informática eran detectadas con mayor facilidad a comparación actual, esto debido a que las redes eran de uso privado y legítimo del ejército militar de los Estados Unidos permitiendo así la descentralización de la información de forma estratégica, sin embargo, con la constante aceleración global tecnológica se ha podido ver como la pequeña red denominada ARPANET correspondientes a las siglas **Advanced Research Projects Agency Network**, siendo la Red de la Agencia de Proyectos de Investigación Avanzada ha dejado de ser exclusiva para un selecto grupo de personas sino que se ha convertido en lo que hoy es llamado Internet.

Así, permitiendo conectar miles y miles de dispositivos que van desde aquellos conectados a redes domésticas u hogares hasta organizaciones que comprenden empresas, gobiernos, infraestructuras críticas, bancos e infinidad de entes que pueden aprovechar este recurso de diferentes formas (Ptolomeo).

Al igual que el internet, las vulnerabilidades que antes eran fáciles de detectar y mitigar en la actualidad se ha convertido en una tarea compleja que comprende software, hardware, tecnologías, productos, versiones, etc.

Dado lo anterior, estas vulnerabilidades son explotadas o aprovechadas por diferentes personas u organizaciones estructuras que pretender tener beneficios monetarios, políticos también conocidos como hacktivismo, ventajas en el mercado, demostración de superioridad o simplemente desafíos personas.

De acuerdo con la compañía internacional Thycotic dedicada a la seguridad informática en una encuesta realizada en Black Hat, la conferencia de seguridad que reúne a cientos de hackers a nivel global, teniendo presente que el término "hacker" no hace referencia a aquella persona dedicada al robo de dinero, diferente a un cibercriminal.

La encuesta expone que el 86% de los hackers están convencidos de que nunca serán castigados por sus acciones y no asumen responsabilidad alguna sobre las consecuencias. Al parecer, la impunidad es la primera llamada a la acción en el cibercrimen (Kaspersky, 2014). El 40% de los encuestados afirma que su primera opción sería los contratistas de la empresa, ya que éstos tienen acceso a las redes corporativas y no están completamente regulados por las políticas de seguridad.

Por otra parte, la encuesta también reveló que el 51% de los hackers suelen realizar ataques por la mera diversión de hacerlo. Sin embargo, el 30% de los encuestados asegura que nunca infringe sus principios éticos.

Finalmente, el 88% de los hackers asegura que, a pesar de contar con grandes capacidades y conocimientos en computación, ni siquiera ellos están exentos de convertirse en víctimas de un ataque o de un robo masivo de datos, perpetrado por otros hackers.

2.2.7.1 Tipos de Vulnerabilidades

Las vulnerabilidades no solo pueden ser lógicas, sino que también se encuentran inmersas en el ambiente físico que van desde aquellas imprevistas generadas por la naturaleza hasta aquellas presentes en el ser humano.

Existen tres tipos de vulnerabilidades que se presentaran a continuación:

- Lógicas
- Físicas
- Humana

2.2.7.1.1 Vulnerabilidades Físicas

Las vulnerabilidades físicas son aquellas que afectan la infraestructura de una organización de manera física, acuñando así la posibilidad de que un activo sufra daños por causas del ambiente o desastres naturales, tales como incendios, terremotos, tormentas, que pueden alterar el correcto funcionamiento de un entorno, y que pueden presentar una denegación en el servicio, una afectación en la disponibilidad y desencadenar problemas mayores.

Así mismo, otra vulnerabilidad física es aquella presente en controles de acceso a organizaciones y diversidad de entornos (3Ciencias, 2018), en donde cualquier persona podría entrar a una empresa y sustraer un equipo de cómputo permitiéndole así conocer información confidencial que puede ser vendida o aprovechada en beneficios personales.

2.2.7.1.2 Vulnerabilidades Lógicas

Las vulnerabilidades lógicas son aquellas que afectan todo aquello que es intangible en la computadora y el desarrollo de las operaciones de estos, siendo estas encontradas en:

- Configuración
- Actualización
- Desarrollo

Las vulnerabilidades de **configuración** son realizadas por defecto por los fabricantes en muchos de los softwares embebidos en los hardware que tienen como objetivo mostrar el correcto funcionamiento de un dispositivo y sus características, incluyendo así firewalls, routers, switches, access point, teléfonos VoIP, servidores, IoT, entre muchos otros, que a la final deben ser cambiadas y gestionadas de forma correcta en ambientes de producción.

Las vulnerabilidades de **actualización**, como se ha venido exponiendo, cada día el nivel de vulnerabilidades de diferentes índoles va en crecimiento, forzando a muchos fabricantes a generar parches o actualizaciones de último momento que permitan mantener los sistemas seguros, así mismo permitiendo corregir fallos en el software o agregando funcionalidades.

Muchos usuarios no se percatan o posponen estas actualizaciones siendo objetivo primordial de cibercriminales que explotan una vulnerabilidad habiendo sido ya corregida por el fabricante (3Ciencias, 2018).

Las vulnerabilidades de **desarrollo** son de las más presentes, muchos de los desarrolladores de software hacen la reutilización de software en proyectos o habitúan utilizar “copiar y pegar” en donde muchas de esas líneas de código no están depuradas de forma óptima o simplemente poseen vulnerabilidades.

Un artículo escrito en el año 2019 titulado “An Empirical Study of C++ Vulnerabilities in Crowd-Sourced Code Examples” (Morteza Verdi, 2019) confirma esta situación, en donde se expone que se revisaron más de 72,000 fragmentos de código C++ tomados de 1,325 publicaciones de Stack Overflow. Encontraron 69 fragmentos vulnerables de 29 tipos diferentes de ataque.

Ahora bien, puede no parecer mucho teniendo en cuenta la gran cantidad de proyectos de GitHub, esos 69 fragmentos vulnerables aparecieron en 2,589 repositorios de GitHub, lo cual es alarmante y preocupante, muchas de las soluciones de software que se puedan estar utilizando en estos puedan contener código “reciclado”.

{A pesar de que los investigadores asumieron la responsabilidad de notificar a los autores de los proyectos afectados de GitHub, solo algunos optaron por corregir las vulnerabilidades que consistían en CWE (enumeración de debilidad común) conocida (TI, 2019), dando un panorama desalentador en donde la respuesta a este tipo de situaciones es atendida por pocos y mientras tanto cibercriminales siguen haciendo de las suyas.

2.2.7.1.3 Vulnerabilidad Humana

En este punto quiero traer a colación la frase de Antonio Ramos quien es presentador y fundador del programa Mundo Hacker, experto en hacking y director de proyectos en stackoverflow, él nos comenta que “las vulnerabilidades humanas son difíciles de erradicar y nos hacen hakeables” (Antonio, 20215).

Como consecuencia, es un punto de partida importante en donde se debe tener presente que el ser humano no cuenta con ningún mecanismo de defensa como firewall o antivirus que permita detectar actores que puedan ser amenazantes en alguna situación, en muchas oportunidades los errores y accidentes que amenazan a la seguridad de la información ocurren de forma descuidada o por ingenuidad, siendo la ingeniería social aliado de algunos actores que lo utilizan a beneficio propio sin necesidad de tener conocimientos técnicos. La principal vulnerabilidad es la falta de capacitación y la falta de conciencia de seguridad para las actividades (Ptolomeo).

2.2.7.2 Clases de Vulnerabilidades

La competitividad del mercado global ha sido uno de los factores que ha impedido de cierta forma la implementación adecuada de seguridad en cada uno de los productos que se encuentran en el mercado actual, en donde se han encontrado vulnerabilidades o fallos de seguridad en dispositivos que menos se imaginan.

Un ejemplo de ello se encuentra en el reportaje realizado por Zoe Kleinman del noticiero BBC en el año 2013 nos muestra un panorama actual, en donde un inodoro automático de marca Satis que tiene un precio de US\$5.686 y que brinda los servicios de tirado de cadena automático, así como controlar desde el teléfono cosas como un spray de agua, música y emisión de fragancias, resulta ser vulnerable a ataques informáticos, por lo cual cualquier teléfono con la aplicación de control de este inodoro podría activarlo, esto debido a que todos los aseos Inax Satis tienen la misma clave (cuatro ceros) (Kleinman, 2013), lo que significa que no se puede resetear y que puede ser activada por cualquier teléfono con la aplicación.

De acuerdo con el reporte elaborado por expertos en seguridad de la firma Trustwave's Spiderlabs, “un atacante podría simplemente descargar la aplicación de My

Satis y usarlo para que el inodoro tire todo el rato de la cadena, lo que aumentaría el uso de agua y los costos para su dueño".

Por lo mostrado anteriormente, es importante que se tome en serio cada aspecto de seguridad en un producto de software o hardware desde su diseño, dando la tranquilidad a usuarios finales y teniendo muy en cuenta que los dispositivos tecnológicos solo traducen simples 1 y 0, en donde el criterio de sensibilidad en los datos debe ser dado por el factor humano (García Rambla, Alonso, & González, 2017).

En el siguiente apartado se darán a conocer algunas vulnerabilidades que pueden ser encontradas en diferentes productos de software y que intentar ser mitigados por la seguridad informática.

2.2.7.2.1 Vulnerabilidad de Diseño

Son aquellas vulnerabilidades producto del diseño ineficiente por parte de un actor responsable a la hora de implementar algún tipo de tecnología, en donde por ejemplo se encuentran administradores de redes, desarrolladores de software o analistas de seguridad informática, se comprenden aspectos como la falta de experiencia o falta de conocimiento siendo consecuente la debilidad en el diseño o implementación de protocolos de red, así mismo, política de seguridad deficientes e inexistentes en donde se ven afectados activos tales como la información (3Ciencias, 2018).

2.2.7.2.2 Vulnerabilidad en Implementación

Se produce cuando hay errores de programación o existencia de "puertas traseras" en los sistemas informáticos producidas por descuidos humanos por parte de un ingeniero de software o fabricantes de tecnología (Víctor Manuel CastellanosBernal, 2019), siendo producto de la reutilización de software encontrado en el internet o proyectos software anteriores.

2.2.7.2.3 Vulnerabilidad de Uso

Asociada a una vulnerabilidad lógica de configuración, la principal causa de la vulnerabilidad de uso es por la configuración deficiente de los sistemas informáticos debido a la falta de experiencia o desconocimiento y falta de sensibilización de los usuarios y de los responsables de informática de una organización (INTEF, n.d.).

Directivos y grupos de inversionistas en una empresa consideran un gasto "innecesario", de bajo impacto o aplicable a muy largo plazo la seguridad informática, utilizando aun tecnología obsoleta en su infraestructura quedando cortos frente a la complejidad de ciberataques, dejando a los responsables de informática con limitación gubernamental de tecnologías de seguridad.

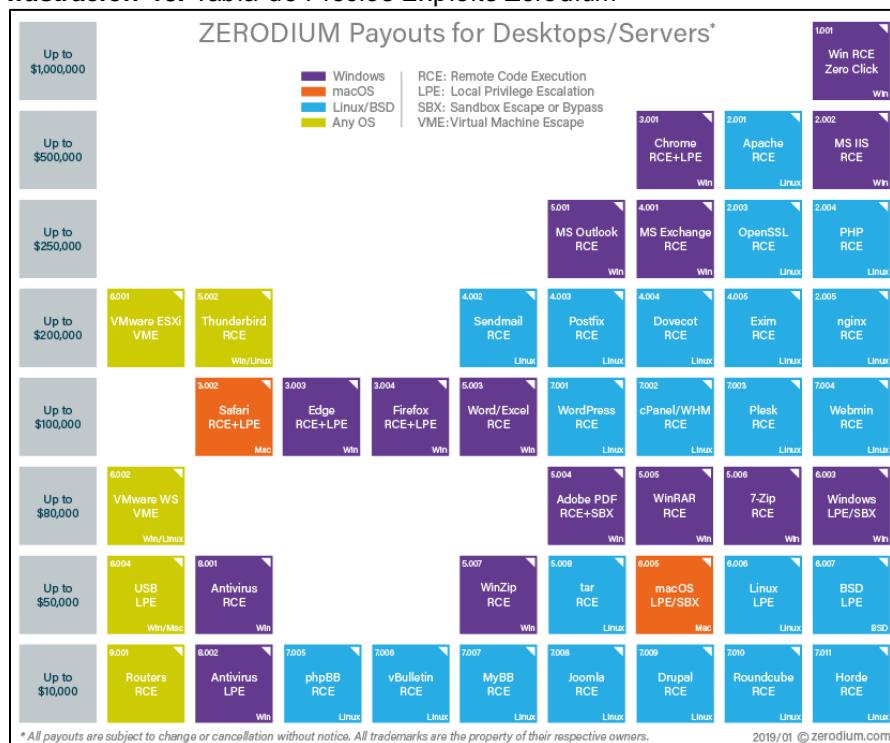
2.2.7.2.4 Vulnerabilidad de Día Cero

Realizadas las correspondientes pruebas de software de un producto por parte de un fabricante, se omiten factores de bajo nivel propensos a brecha de seguridad en donde no existe una solución "conocida" para una vulnerabilidad, sin embargo, se hace público su proceso de explotación, siendo esta una carrera contrarreloj entre el fabricante y el cibercriminal.

Zerodium es una empresa estadounidense dedicada a la seguridad informática fundada en 2015, adicionalmente se dedica a la compra y venta de exploits o códigos de software especialmente codificados para explotar vulnerabilidades de día cero de todos los tipos de tecnología, entre sus clientes se encuentran agencias de gobierno y grandes empresas.

En su página oficial los exploits elegibles de día cero oscilan entre US\$2.000 y US\$2.500.000 por envío. Los montos pagados por Zerodium a los investigadores para adquirir sus exploits originales de día cero o denominados también zero-day dependen de la popularidad y el nivel de seguridad del software/sistema afectado, así como de la calidad del exploit presentado (cadena total o parcial, versiones/sistemas/arquitecturas compatibles), fiabilidad, mitigaciones de exploits omitidas, componentes predeterminados frente a componentes no predeterminados, continuación del proceso, etc.) (ZERODIUM, n.d.). En la Ilustración 18 se observa el precio pagado por exploits de acuerdo a su funcionalidad y tecnología que vulnera.

Ilustración 18. Tabla de Precios Exploits Zerodium



Fuente: Tomado de <https://bit.ly/3eDrsiB>

De acuerdo con el portal de noticias de seguridad informática, existe un mercado subterráneo de herramientas para “hackear” a cualquiera. La Deep Web está llena de mercados negros donde se pueden encargar hacks dirigidos a personas o empresas en particular.

Hay lugares en los que se comparten exploits y descubrimientos informáticos de toda índole. Hasta ahora ese espacio estaba reservado sólo a crackers, como no podía ser de otra manera. Ahora las tornas podrían haber cambiado. Podría haber actores externos a este submundo deseando beneficiarse de él, consiguiendo convertirse en algo que podríamos definir como traficantes corporativos de hacks.

Como cabría esperar, desde la compañía se tiene otro punto de vista. Según ellos, lo que pretenden es ayudar a las fuerzas del orden a investigar con mejores herramientas a su disposición.

La historia entre el FBI y Apple muestra el aspecto más interesante del negocio de los zero-day, que es la necesidad de que las agencias gubernamentales accedan a fallos sin parchear para llevar a cabo investigaciones y salvar vidas (Chaouki Bekrar, CEO de Zerodium), siendo este un arma de doble filo, utilizado para la vigilancia y perdida de la privacidad en el entorno digital, y que se traduce en gobiernos exigiendo a las empresas que otorguen acceso ilimitado a cualquier dispositivo a través de una puerta trasera, algo que el FBI ya pidió a Apple a raíz de los incidentes de San Bernardino (Martínez, 2016).

2.2.7.2.5 Vulnerabilidad de Desbordamiento de Buffer

Ocurre cuando el programador no controla la cantidad de datos en memoria que se copian en el buffer de un programa, de forma que si esa cantidad es superior a la capacidad del buffer los bytes sobrantes se almacenan en zonas de memoria adyacentes, sobrescribiendo su contenido original.

Se puede aprovechar por parte de un cibercriminal para introducir su propio código en este espacio de memoria para ejecutar cualquier otra tarea (OWASP, OWASP Buffer Overflow Vulnerabilities, n.d.), por ejemplo, va desde abrir un simple programa hasta injectar payloads - carga útil que se ejecuta luego de explotar una vulnerabilidad - en los cuales se introduce cierta cantidad de memoria o inclusive dentro de los backdoor o puerta trasera permitiendo ejecutar código con privilegios de administrador.

2.2.7.2.6 Vulnerabilidad de Cross Site Scripting (XSS)

La vulnerabilidad de ejecución de comandos en sitio cruzado o también conocida como Cross Site Scripting (XSS) es la segunda vulnerabilidad más frecuente en OWASP Top 10, y se encuentra en alrededor de dos tercios de todas las aplicaciones web, son un tipo de inyección, en el que los scripts tipo VBScript o JavaScript maliciosos se injetan en sitios web benignos y confiables (INTEF, n.d.). Ocurre cuando un atacante usa una aplicación web para enviar código malicioso, generalmente en forma de script del lado del navegador, a un usuario final diferente.

Un atacante puede usar XSS para enviar un script malicioso a un usuario desprevenido. El navegador del usuario final no tiene forma de saber que no se debe confiar en el script y ejecutará el script (KirstenS, n.d.). Debido a que cree que el script provino de una fuente confiable, el script malicioso puede acceder a cualquier cookie, tokens de sesión u otra información confidencial retenida por el navegador y utilizada con ese sitio.

2.2.7.2.7 Vulnerabilidad de Inyección SQL

De acuerdo con OWASP, consiste en la inserción o "inyección" de una consulta SQL a través de los datos de entrada del cliente a la aplicación. Una explotación de inyección SQL exitosa puede leer datos confidenciales de la base de datos, modificar datos de la base de datos (Insertar/Actualizar/Eliminar), ejecutar operaciones de administración en la base de datos (como cerrar el DBMS), recuperar el contenido de un archivo dado presente en el archivo DBMS sistema y en algunos casos emitir comandos al sistema operativo (OWASP, OWASP SQL Injection, n.d.).

La vulnerabilidad de inyección SQL son un tipo de ataque de inyección, en el que los comandos SQL se inyectan en la entrada del plano de datos para efectuar la ejecución de comandos SQL predefinidos.

2.2.7.2.8 Vulnerabilidad de Denegación de Servicio

La vulnerabilidad de denegación de servicio o Denial of Service (DoS) se centra en hacer que un recurso (sitio, aplicación, servidor) no esté disponible para el propósito para el que fue diseñado. Existen diversas formas de hacer que un servicio no esté disponible para usuarios legítimos mediante la manipulación de paquetes de red, programación, vulnerabilidades lógicas o de manejo de recursos, entre otros (OWASP, n.d.).

Si un servicio recibe una gran cantidad de solicitudes, puede dejar de estar disponible para usuarios legítimos. Del mismo modo, un servicio puede detenerse si se aprovecha una vulnerabilidad de programación o la forma en que el servicio maneja los recursos que utiliza.

En algunas oportunidades dependiendo del objetivo de un atacante, puede inyectar y ejecutar código arbitrario mientras realiza un ataque DoS para acceder a información crítica o ejecutar comandos en el servidor.

2.2.7.2.9 Vulnerabilidad de Contraseñas en Blanco o Débiles

Las contraseñas son una característica de seguridad de uso habitual en diferentes sistemas para confirmar la identidad de un usuario, pero aun así presenta problemas. Los inicios de sesión a menudo tienen contraseñas que se dejan en blanco intencionalmente, incluso cuentas de administración (Nilsson & Virta, 2006). Los gusanos y los programas de pirateo a menudo verifican estas condiciones, una contraseña en blanco o una contraseña que sea la misma que la del inicio de sesión. Las contraseñas deben cambiarse regularmente de acuerdo con ciertos requisitos.

2.2.7.3 Registro de Vulnerabilidades (CVE)

A nivel global existe una lista con información registrada acerca de vulnerabilidades de seguridad informática encontradas y reportadas por investigadores, esta lista es conocida como Vulnerabilidades y Exposiciones Comunes o denominada en lenguaje inglés como Common Vulnerabilities and Exposures (CVE).

Los CVE se pusieron en marcha en 1999, cuando la mayoría de las herramientas de ciberseguridad utilizaban sus propias bases de datos con sus propios nombres para detectar vulnerabilidades en materia de seguridad. En ese momento no había ninguna variación significativa entre los productos y no era fácil determinar cuándo las diferentes bases de datos se referían al mismo problema (Mitre, n.d.). Las consecuencias fueron posibles lagunas en la cobertura de seguridad y la falta de interoperabilidad efectiva entre las diferentes bases de datos y herramientas.

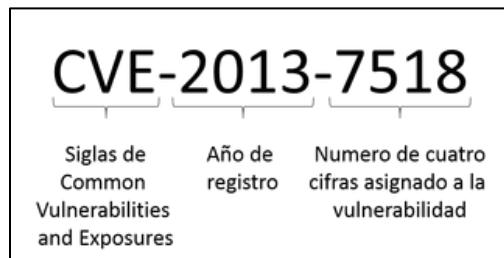
En la actualidad MITRE Corporation se encarga de supervisar los CVE con el financiamiento de la Agencia de Seguridad de Infraestructura y Ciberseguridad, que forma parte del Departamento de Seguridad Nacional de Estados Unidos (RedHat, n.d.).

Siendo CVE ahora el estándar de la industria para nombres de vulnerabilidad. Los identificadores CVE proporcionan puntos de referencia para el intercambio de datos de modo que los productos y servicios de seguridad cibernética puedan comunicarse entre sí.

También proporcionan una base de referencia para evaluar la cobertura de los instrumentos y servicios a fin de que los usuarios puedan determinar qué instrumentos son más eficaces y apropiados para las necesidades de su organización. En resumen, los productos y servicios compatibles con CVE ofrecen una mejor cobertura y una interoperabilidad más fácil y mejorar la seguridad.

Ahora bien, las autoridades de numeración de CVE (CNA) - que afectan a productos dentro de su alcance distinto y acordado - son las encargadas de asignar los números de identificación de CVE (CVE-ID) y está formado por las siglas de este diccionario seguidas por el año en que es registrada la vulnerabilidad o exposición y un número arbitrario de cuatro dígitos. Estos tres elementos van separados por un guion resultando un identificador como se puede observar en la Ilustración 19.

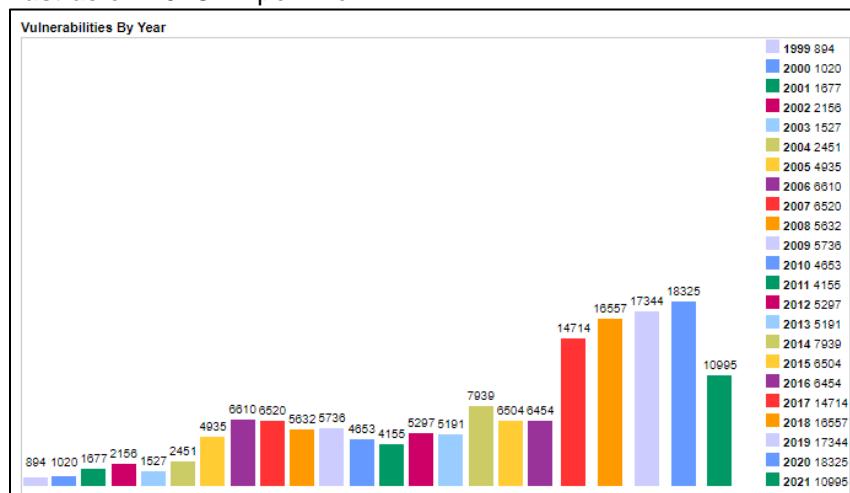
Ilustración 19. Numeración de CVE



Fuente: Tomado de <https://bit.ly/3avzs5>

Este formato se ha mantenido durante mucho tiempo, pero en el año 2014 ha cambiado y se evidencia un fenómeno alarmante. Revisando el historial del total de vulnerabilidades reportadas por año en la página web CVE Details (ver Ilustración 20) ha crecido de forma precipitosa, donde solamente en el año 2020 se reportaron 18325 vulnerabilidades, parece que se han quedado corto los 9.999 números para un año (Schiavo, 2014). Así que desde el año 2014 este identificador puede contener hasta siete (7) números para asignación de una vulnerabilidad, que superan la barrera del 9.999.

Ilustración 20. CVE por Año



Fuente: Tomado de <https://bit.ly/2xTPQfc>

2.2.7.4 Sistema de Puntuación de Vulnerabilidad (CVSS)

Actualmente, los departamentos de informática, encargados de la gestión de los activos de TI deben determinar y evaluar las vulnerabilidades en una amplia variedad de sistemas, incluyendo así hardware y plataformas de software (AndalucíaCERT, 2014). Es por ello que se hace necesario poder establecer una prioridad según un nivel de riesgo con la meta de evaluar qué vulnerabilidades son potencialmente más peligrosas y, por tanto, más urgentes de remediar.

El Common Vulnerability Scoring System (CVSS) o denominado Sistema de Calificación de Vulnerabilidades, es un marco abierto en su versión actual 3.1 que categoriza las principales características técnicas de las vulnerabilidades de software, hardware y firmware (ORG F. , FIRST ORG, n.d.).

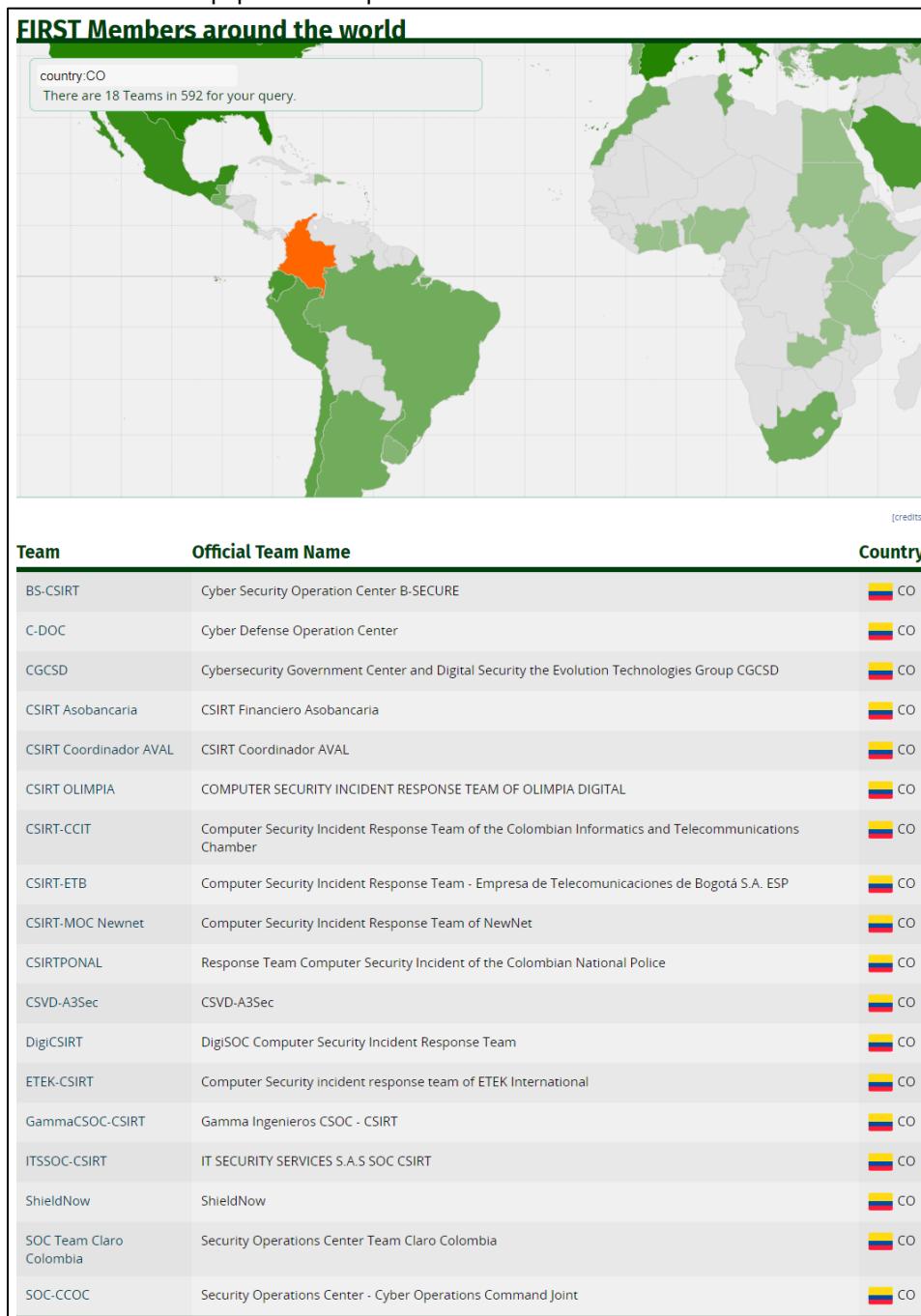
Como propósito principal se centra en obtener un valor medible que ofrezca una idea del peligro potencial que supone la presencia de una vulnerabilidad determinada en algún producto tecnológico, ayudando a transmitir y evaluar la gravedad y permitiendo determinar la urgencia y prioridad de esta.

CVSS es propiedad de FIRST.Org, Inc. (FIRST), una organización sin fines de lucro con sede en Estados Unidos, dando así sus primeros pasos en 1990 con la motivación de cubrir un vacío en lo que ha respuesta a incidentes de seguridad se refiere. Esta insuficiencia puso de manifiesto con la primera muestra relevante de un gusano de internet conocido como Morris, creado en el MIT y que produjo importantes daños al expandirse por internet de forma incontrolada (López, 2015).

Por ello, este incidente dio lugar al nacimiento del Forum of Incident Response and Security Teams (FIRST), que desde entonces se ha posicionado y reconocido como la organización líder en la gestión de respuesta a incidentes, cuya misión es ayudar a los equipos de respuesta a incidentes de seguridad informática en todo el mundo.

A nivel global FIRST cuenta con la colaboración de equipos de respuesta a incidentes que incluyen un margen amplio de sectores como el de la educación, telecomunicaciones, comercio y fabricantes, gobiernos y entidades militares (ORG F. , FIRST ORG, n.d.). En la actualidad cuenta con un total de 592 equipos en 98 países diferentes, en donde Colombia posee 18 equipos de respuesta a incidentes como se puede observar en la Ilustración 21, siendo de los más grandes de Latinoamérica.

Ilustración 21. Equipos de Respuesta de Incidentes en Colombia



Fuente: Tomado de <https://bit.ly/2YbDPwH>

Es importante resaltar que ninguna organización “posee” CVSS, no siendo necesario pertenecer a FIRST para utilizar o emplear CVSS. La única petición de FIRST es que las organizaciones que publiquen los resultados se ajusten a las directrices descritas en la guía de referencia de CVSS y aporten información de cómo se obtuvo (AndalucíaCERT, 2014).

2.2.7.5 Bases de Datos de Vulnerabilidades

Frente al crecimiento de vulnerabilidades a través de los años, ha sido fundamental contar con plataformas que permitan almacenar, mantener y difundir información sobre vulnerabilidades de seguridad informática que se descubren a diario,

todo ello para tener un panorama claro y certero de las diferentes amenazas presentes en infraestructuras tecnológicas (Kyriakos Kritikos, 2019).

La mayoría de las Vulnerability Databases (Vdbs) no se centran en cualquier tipo de vulnerabilidad; muy pocos tienen una limitante. Para la evaluación de riesgos, muchos Vdbs adoptan Sistema de puntuación de vulnerabilidad (CVSS), mientras que muy pocos sólo proporcionan un impacto.

Algunos Vdbs proporcionan tanto información de impacto como de riesgo para cada vulnerabilidad (Vasilis Katos, 2019). También podrían incluir información extra, como qué artefacto es afectado y cómo la vulnerabilidad se podría abordar. Muchos Vdbs también se ajustan a las normas que tipo de identificación normalizada o determinación de la información para cubrirse. En la Tabla 6 se muestran las diferentes bases de datos disponibles para la validación y verificación de vulnerabilidades.

Tabla 6. Bases de Datos de Vulnerabilidades

Base de Datos	Tipo de DDBB	Descripción
NVD	Datos CVE	https://nvd.nist.gov/ El NVD es el repositorio de vulnerabilidades basada en estándares del gobierno de los Estados Unidos. El NVD incluye bases de datos de referencias de listas de verificación de seguridad, fallas de software relacionadas con la seguridad, configuraciones incorrectas, nombres de productos y métricas de impacto.
ATT&CK	Patrones del atacante (técnicas y tácticas)	https://attack.mitre.org/ MITRE ATT&CK™ es una base de conocimiento, globalmente accesible de tácticas y técnicas adversas basadas en observaciones del mundo real.
Shodan	Número de exploits	https://www.shodan.io/ Base de datos de dispositivos conectados a Internet (por ejemplo, cámaras web, routers, servidores, etc.) que adquieren datos de varios puertos (por ejemplo, HTTP/ HTTPS: puerto 80, 8080, 443, 8443).
Exploit Database	Datos no CVE	https://www.exploit-db.com/about-exploit-db contiene información sobre exploits públicos y el correspondiente software vulnerable. La colección de exploits se adquiere de envíos directos, listas de correo y otras fuentes públicas.
CVE details	Datos CVE	https://cve.mitre.org/ CVE®, es una base de datos que contiene detalles de vulnerabilidades de seguridad cibernética conocidas públicamente, incluido un número de identificación, una descripción y al menos una referencia pública.
Zero Day Initiative	CVE y no CVE	https://www.zerodayinitiative.com/ fomenta la notificación privada de vulnerabilidades de día cero a los proveedores afectados por parte de los investigadores que premian financieramente (un programa de recompensas de errores

		independiente del proveedor). No se hacen públicos detalles técnicos sobre vulnerabilidades individuales hasta después de que el proveedor haya lanzado parches. ZDI no revende ni redistribuye las vulnerabilidades.
ThreatConnect	Número de incidentes relacionados con CVE	https://threatconnect.com/ Inteligencia de amenazas automatizada para sistemas Intel.
VulDB	Precios de exploits y categorías de software	https://vuldb.com/ base de datos de vulnerabilidad que documenta y explica vulnerabilidades y exploits.
US CERT	Sector industrial	https://www.us-cert.gov/ La Agencia de Seguridad Cibernética y Seguridad de Infraestructuras (CISA) del Departamento de Seguridad Interior de los Estados Unidos tiene como objetivo mejorar la seguridad, resiliencia y fiabilidad de la infraestructura de seguridad cibernética y comunicaciones de los Estados Unidos.
Zerodium	Recompensa por errores y precio a exploits	https://zerodium.com/ Una plataforma de adquisición de zero-day. Fundada por expertos en seguridad cibernética con experiencia en investigación avanzada de vulnerabilidades.

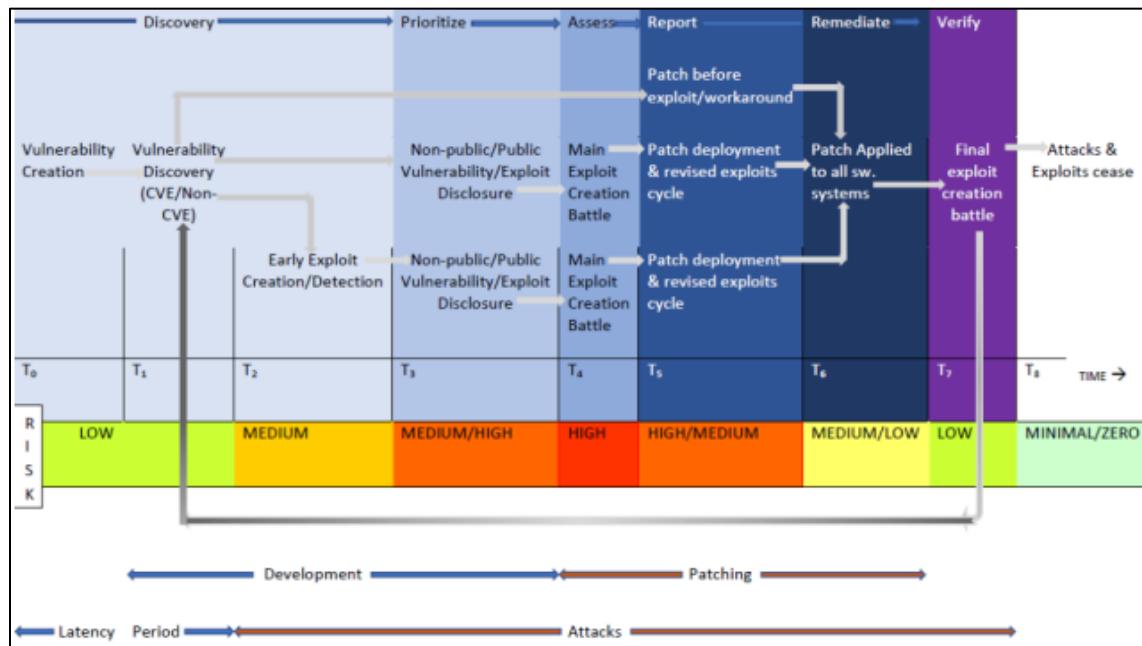
Fuente: Tomado de <https://bit.ly/2KTLSXb>

2.2.7.6 Ciclo de Vida de una Vulnerabilidad

Una vulnerabilidad por sí misma no representa la materialización de un ataque, únicamente representa un riesgo al que se encuentra expuesto el activo o proceso (Víctor Manuel CastellanosBernal, 2019), sin embargo, las vulnerabilidades informáticas no son un tema que se deban tomar a la ligera, en donde cada aspecto es importante para poder llevar a cabo una remediación de un sistema de forma adecuada sin verse afectada ninguna parte de la triada de la seguridad compuesta por la confidencialidad, integridad y disponibilidad, manteniendo seguro cada activo informático corporativo esencial.

Al igual que muchos procesos encontrados en la industria manufacturera compuestos por un ciclo de vida útil de un producto, las vulnerabilidades de igual forma cuentan con un ciclo de vida propuesto por Arbaugh Et Al en el año 2000 como se muestra en la Ilustración 22, en esta se exhibe un gráfico idealizado del ciclo de explotación de vulnerabilidades.

Ilustración 22. Ciclo de Vida de una Vulnerabilidad



Fuente: Tomado de <https://bit.ly/2xRzWlx>

La imagen anterior es completa, proporcionada por Enisa en su informe denominado “State Vulnerabilities 2018/2019” (Vasilis Katos, 2019), en donde modela lo propuesto por Arbaugh Et Al y se evidencia desde el momento que es descubierta una vulnerabilidad el posible riesgo que pueda representar mientras transcurre su tiempo de reporte/reparación y finalmente la liberación del parche correspondiente.

Pero, en muchas organizaciones este ciclo de vida no se cierra por parte de los responsables de informática y queda inconcluso en el “T6” denominado “Remediación”, a pesar que los fabricantes de la tecnología hallan liberado el parche o actualización correspondiente para cerrar una vulnerabilidad hace meses, inclusive hace años, permaneciendo el activo de la información en un riesgo medio/alto que puede convertirse en un riesgo alto/muy alto de haber un actor interesado si llegase descubrir y explotar la vulnerabilidad.

Según Panda Security (Security P. , Panda Security, 2018), los ataques de malware y ransomware se encuentran en las consecuencias más llamativas de una vulnerabilidad sin parchear, pero no las únicas. Algunos de los casos más serios de exfiltraciones de datos personales han sido posibles gracias a sistemas informáticos sin parchear.

En 2017, la empresa estadounidense Equifax reveló que había perdido los datos personales de más de 145 millones de personas, en una de las mayores brechas de este tipo en la historia. Todo debido a una vulnerabilidad en Apache Struts que se aprovechó para Cerber. Según Equifax, la responsabilidad fue de un empleado que no aplicó el parche relevante, un parche disponible dos meses antes de la brecha y que podría haberla mitigado.

El caso no es único. La aseguradora Nationwide Mutual Insurance acordó pagar 5,5 millones de dólares por una brecha de los datos de 1,27 millones de personas en 2012, que también fue facilitada por una vulnerabilidad en una aplicación Web para la que existía un parche tres años antes del incidente.

De hecho, según un estudio, más del 80% de las brechas de datos personales son el resultado de una mala gestión de los parches. Esto quiere decir que una empresa puede reducir de manera significante el riesgo de sufrir un incidente de este tipo aplicando una política eficaz de parches. Siendo las excusas más relevantes una falta de recursos y tiempo por parte de responsable de informática en una organización para buscar y aplicar parches.

2.2.7.7 Detección de Vulnerabilidades

Diseñar enfoques efectivos para detectar vulnerabilidades es importante para los evaluadores de la seguridad de la información (David A. Franco, 2013), siendo así que las vulnerabilidades pueden ser encontradas mediante herramientas, que permiten realizar un escaneo de puertos con la meta de verificar cuales están abiertos para intentar obtener información sobre el servicio que se encuentre ejecutando en ese momento y con esta información buscar vulnerabilidades asociadas precisamente a esos servicios (3Ciencias, 2018). Se tienen dos modos de detectarse.

- Escáner de vulnerabilidades automáticos
- Análisis manuales

En la forma de **escáner de vulnerabilidades automáticos** entre las herramientas más importantes se pueden mencionar NeXpose, Acunetix, W3af, Nikto, NMAP, Nessus, OpenVas, entre otros. Un escáner de vulnerabilidades comienza como un escáner de puertos e intenta identificar todos los hosts que se ejecutan en el rango de IP definido. Cuando se encuentran los hosts, el escáner intenta encontrar todos los puertos abiertos y los servicios correspondientes en todos los hosts activos.

En la mayoría de los escáneres existe la posibilidad de configurar diferentes modos de escaneo y también establecer qué puertos escanear (Nilsson & Virta, 2006). El objetivo del escáner es identificar las vulnerabilidades en el host escaneado y esto se hace mediante la comparación de los sistemas operativos en ejecución y las aplicaciones de software que se ejecutan con vulnerabilidades conocidas almacenadas en una base de datos.

Incluso con los albores de los programas de aprendizaje automático, todavía hay elementos que requieren atención humana a los detalles como en el **análisis manual**, para determinar con precisión o verificar vulnerabilidades presentes. Aquí es donde el valor de un auditor es tan importante.

Los auditores de seguridad informática o penetration tester avanzados pueden usar su ingenio, lógica empresarial, experiencia y habilidades en el análisis para descubrir las fallas profundas y anidadas dentro de un sistema. Si una organización solo contrata a una empresa que utiliza escáneres automáticos de vulnerabilidades, podrían perderse elementos críticos (Burgett, 2019).

2.2.7.8 Explotación de Vulnerabilidades

Todos los días se escucha, se lee, se comenta sobre los riesgos que representan esos seres “extraños” para los sistemas de información, para la privacidad personal e industrial. Se asocian los riesgos a factores externos, a “hackers” que desde afuera vulneran la seguridad de sistemas (Néstor Dabío Duque Méndez).

De acuerdo con Eirc Raymon (Raymond, 2001), el término hacker en su mayoría tiene que ver con la habilidad técnica y el deleite en resolver problemas y superar los límites.

De igual forma, la Real Academia Española (RAE) (RAE, 2019), en su segunda acepción, establece que es una “persona experta en el manejo de computadoras, que se ocupa de la seguridad de los sistemas y de desarrollar técnicas de mejora”.

Existe una comunidad, una cultura compartida, de programadores expertos y asistentes de redes que remontan su historia a través de décadas hasta los primeros minicomputadores de tiempo compartido y los primeros experimentos ARPAnet (Raymond, 2001). Los miembros de esta cultura originaron el término 'hacker'. Los hackers construyeron Internet. Los hackers hicieron del sistema operativo Unix lo que es hoy.

Los hackers hacen que la World Wide Web funcione. Pero, este término ha sufrido cambios culturales con el pasar de los años a causa de las películas y noticias que lo asocian a crackers que dañan sistemas y roban información. La diferencia básica está en que: los hackers construyen cosas, los crackers las rompen.

Ahora bien, en el mundo del hacking existen diversos tipos de hackers, todos ellos con diferentes intenciones y catalogados gracias a las películas de vaqueros antiguas en donde el personaje bueno tenía el sombrero blanco y el malvado un sombrero negro (LASKOW, 2017). En general todos lo hacker aprovechan las vulnerabilidades sin embargo la fuerte diferencia es la finalidad de su uso, en donde:

Un '**Black Hat**' Hackers o “crackers” es un individuo que intenta lograr un ingreso no autorizado en un sistema o red para explotarlos por razones maliciosas. El hacker de sombrero negro no tiene ninguna autorización o autoridad para comprometer sus objetivos. Intentan infilir daños al comprometer los sistemas de seguridad, alterar las funciones de los sitios web y las redes, o apagar los sistemas (EC-Council, 2018). Por lo general lo hacen para robar u obtener acceso a contraseñas, información financiera y otros datos personales.

Los '**White Hat**' Hackers eligen usar sus “poderes” para el bien en lugar del mal. También conocidos como "hackers éticos" o "ethical hackers". Los hackers de sombrero blanco en algunas oportunidades pueden ser empleados remunerados o contratistas que trabajan para una organización como especialistas en seguridad que intentan encontrar vulnerabilidades a través de la piratería.

Los piratas informáticos de sombrero blanco emplean los mismos métodos de pirateo que los sombreros negros, con una excepción; lo hacen en primera instancia con la autorización del propietario del sistema, lo que hace que el proceso sea completamente legal (Security N. , n.d.). Los hackers de sombrero blanco realizan pruebas de penetración, prueban los sistemas de seguridad en el lugar y realizan evaluaciones de vulnerabilidad para las empresas.

Adicionalmente, como en la vida, hay áreas grises que no son negras ni blancas. Los “**Grey Hat**” Hackers son una combinación de actividades de sombrero negro y sombrero blanco. A menudo, los piratas informáticos de sombrero gris buscarán vulnerabilidades en un sistema sin el permiso o conocimiento del propietario (Security N. , n.d.). Si se encuentran problemas, los informarán al propietario y, en algunas ocasiones,

solicitarán una pequeña precio para remediar el problema. Si el propietario no responde o no cumple, a veces los piratas informáticos publicarán el exploit recientemente encontrado en internet para que el mundo lo vea.

A pesar que la palabra hacker tiende a despertar connotaciones negativas cuando se hace referencia a ella, es significativo recordar que no todos los hackers son iguales. Si no tuviéramos hackers de sombrero blanco buscando diligentemente amenazas y vulnerabilidades antes de que los sombreros negros puedan encontrarlos (Security N. , n.d.), entonces seguramente habría mucha más actividad que involucrara a los ciberdelincuentes que explotan vulnerabilidades y recopilan datos confidenciales de lo que hay ahora.

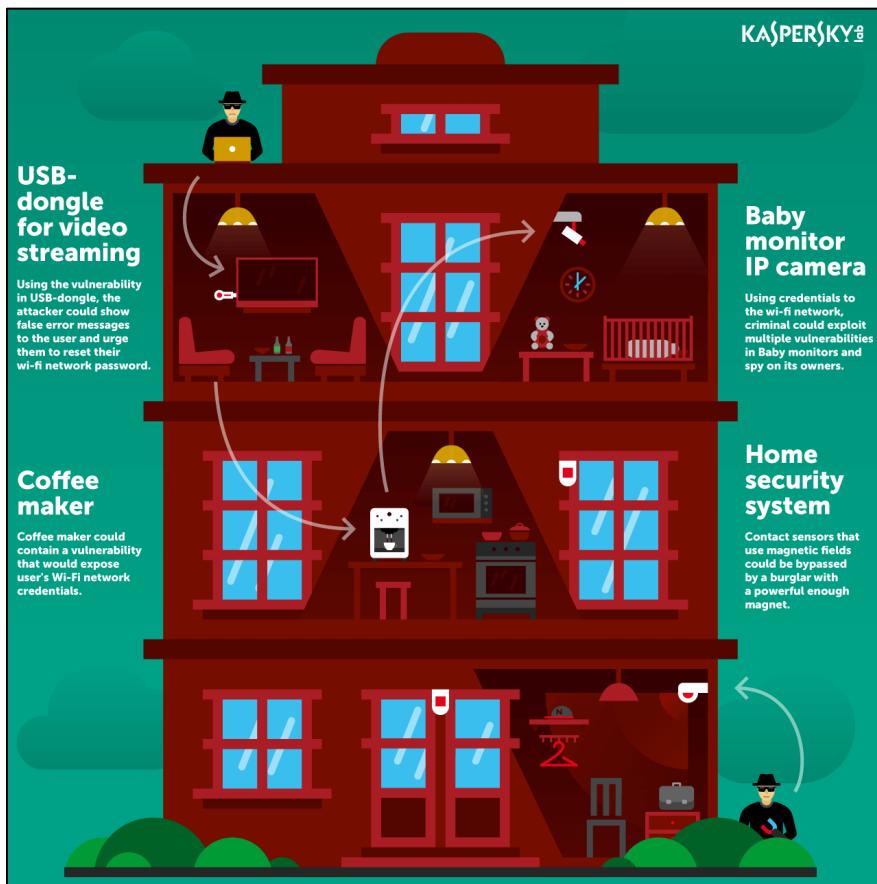
2.2.8 Vulnerabilidades en IoT

Como se ha venido mostrando, las vulnerabilidades en diversos ambientes tecnológicos se encuentran a la orden del día, siendo así que el Internet de las Cosas (IoT) no es la excepción y acompañado del alto crecimiento de estos dispositivos en entornos hogareños y empresariales, hacen esta tecnología un punto crítico que muy pocos le dan importancia.

En 2015 Kaspersky Lab genero un artículo denominado “Más conectado, menos seguro: cómo investigamos IoT para detectar vulnerabilidades” (robot, 2015) en donde se simulaba un entorno hogareño en el cual se hacían uso de dispositivos populares de Internet de las cosas, como Google Chromecast (un dongle USB para transmisión de video), una cámara IP , una máquina de café inteligente y un sistema de seguridad para el hogar como se muestra en la Ilustración 23, todo podía ser controlado por un teléfono inteligente o una aplicación móvil. Los modelos y dispositivos fueron elegidos al azar y eran bastante independientes del vendedor.

El resultado, todos los dispositivos IoT utilizados en la prueba tenían vulnerabilidades y eran pirateables o podrían ser fácilmente comprometidos y utilizados para hacer una orden de pirata informático.

Ilustración 23. Simulación Entorno IoT



Fuente: Tomado de <https://bit.ly/2A2nnoq>

Lo más preocupante sucede con la cámara IP, la cual era un monitor de bebes en donde los investigadores de Kaspersky lograron obtener el control total sobre la cámara: permitiendo así que alguien vea y escuche todo lo que sucede en una habitación, reproduzca un archivo de audio arbitrario en el dispositivo u obtenga acceso de root (administrador) y modifique el software de la cámara, lo que significa que convertirse en el único gobernante de esta pequeña cosa "inteligente" (robot, 2015), dejando la seguridad y privacidad en estos entornos solo en la percepción más no la realidad.

Ahora bien, para entender a qué vulnerabilidades se enfrentan las tecnologías de IoT, el Open Web Application Security Project conocido como OWASP, organización sin ánimo de lucro fundada en el año 2001, que trabaja para mejorar la seguridad del software (ORG O., OWASP ORG, 2001), todo ello a través de proyectos de software de código abierto.

Entre los que se encuentra el tan conocido OWASP Top 10 (ORG O., OWASP Top Ten, 2013) que es un documento de conocimiento estándar para desarrolladores y seguridad de aplicaciones web, de igual forma la guía de pruebas OWASP en su versión oficial 4.0 (ORG O., OWASP Web Security Testing Guide, 2004) que se convierte en un recurso de pruebas de ciberseguridad para desarrolladores de aplicaciones web y profesionales de la seguridad.

Además de definir metodologías que ayuden a la seguridad en entornos Web ha ampliado sus horizontes a tecnologías móviles y como no a entornos IoT en el que se basa el software desarrollado. Este proyecto de OWASP está diseñado para ayudar a los fabricantes, desarrolladores y consumidores a comprender mejor los problemas de

seguridad asociados con Internet of Things, y para permitir a los usuarios en cualquier contexto tomar mejores decisiones de seguridad al crear, implementar o evaluar tecnologías de IoT (ORG O., OWASP Internet of Things, 2018).

El marco OWASP Internet of Things muestra diez cosas elementos que se deben evitar al crear, implementar o administrar sistemas de IoT que se muestra en la Tabla 7.

Tabla 7. OWASP Vulnerabilidades IoT 2018

Ítem	Vulnerabilidad	Descripción
1	Contraseñas débiles, adivinables o codificadas	Uso de credenciales fácilmente adivinables, disponibles públicamente o que no se pueden cambiar, incluidas las puertas traseras en el firmware o el software del cliente que otorgan acceso no autorizado a los sistemas implementados.
2	Servicios de red inseguros	Servicios de red innecesarios o inseguros que se ejecutan en el propio dispositivo, especialmente aquellos expuestos a Internet, que comprometen la confidencialidad, integridad/autenticidad o disponibilidad de información o permiten el control remoto no autorizado.
3	Interfaces inseguras del ecosistema	Web insegura, API de back-end, nube o interfaces móviles en el ecosistema fuera del dispositivo que permite comprometer el dispositivo o sus componentes relacionados. Los problemas comunes incluyen la falta de autenticación/autorización, la falta de cifrado o la debilidad, y la falta de filtrado de entrada y salida.
4	Falta de mecanismo de actualización segura	Falta de capacidad para actualizar de forma segura el dispositivo. Esto incluye la falta de validación de firmware en el dispositivo, la falta de entrega segura (sin cifrar en tránsito), la falta de mecanismos antirretroceso y la falta de notificaciones de cambios de seguridad debido a actualizaciones.
5	Uso de componentes/inseguros u obsoletos	Uso de componentes/bibliotecas de software obsoletos o inseguros que podrían permitir que el dispositivo se vea comprometido. Esto incluye la personalización insegura de las plataformas del sistema operativo y el uso de componentes de software o hardware de terceros de una cadena de suministro comprometida.
6	Protección de privacidad insuficiente	La información personal del usuario almacenada en el dispositivo o en el ecosistema que se utiliza de forma insegura, inadecuada o sin permiso.
7	Transferencia y almacenamiento de datos inseguros	Falta de cifrado o control de acceso de datos confidenciales en cualquier parte del ecosistema, incluso en reposo, en tránsito o durante el procesamiento
8	Falta de gestión de dispositivos	Falta de soporte de seguridad en dispositivos implementados en producción, incluyendo gestión de activos, gestión de actualizaciones, desmantelamiento seguro, monitoreo de sistemas y capacidades de respuesta.

9	Configuración predeterminada insegura	Los dispositivos o sistemas enviados con configuraciones predeterminadas inseguras o carecen de la capacidad de hacer que el sistema sea más seguro al restringir a los operadores la modificación de las configuraciones.
10	Falta de endurecimiento físico	Falta de medidas de endurecimiento físico, lo que permite a los atacantes potenciales obtener información confidencial que puede ayudar en un futuro ataque remoto o tomar el control local del dispositivo.

Fuente: Tomado de <https://bit.ly/3dGErik>

Adicionalmente, OWASP pone a disposición un firmware llamado “IoTGoat” (IoTGoat, 2020) que es deliberadamente inseguro basado en OpenWrt y es utilizado como una plataforma para educar a los desarrolladores de software y profesionales de seguridad con pruebas de vulnerabilidades comúnmente encontradas en dispositivos IoT.

Otra organización que se ha creado viendo la tendencia de IoT y las consecuencias a largo plazo en la no atención de vulnerabilidades, es la IoT Security Fundation, conformada por profesionales de la tecnología y expertos en seguridad, el 23 de septiembre de 2015, se lanzó la Internet of Things Security Foundation (Foundation, The Internet of Things Security Foundation, 2015), quienes al igual que OWASP buscan la adopción segura de soluciones de IoT. Esta organización cuenta con un documento llamado “IoT Security Compliance Framework” que está destinado a ser utilizado por profesionales que diseñan, especifican y adquieren productos relacionados con IoT al proporcionar orientación.

El marco IoT Security Compliance Framework establece un conjunto completo de requisitos de seguridad para aspectos de la organización y el producto (Foundation, The Internet of Things Security Foundation, 2020). Este parámetro es fundamental a la hora de combinarse con el software ya que es posible de acuerdo a la clasificación determinar el riesgo que presenta un dispositivo en un entorno.

Para hacer que el marco sea práctico, ha adoptado un enfoque basado en el riesgo derivado de la tríada CIA de uso común. Si bien no es un modelo perfecto, su simplicidad es su fortaleza y las buenas prácticas de seguridad se pueden derivar de los principios básicos.

Para aplicar un nivel apropiado de cumplimiento de seguridad a un producto, los requisitos de la lista de verificación se clasifican utilizando las siguientes clases de cumplimiento:

- **Clase 0:** donde el compromiso de los datos generados o la pérdida de control probablemente resulte en un impacto poco perceptible en un individuo u organización.
- **Clase 1:** donde es probable que el compromiso de los datos generados o la pérdida de control no tenga más que un impacto limitado en un individuo u organización.
- **Clase 2:** además de la clase 1, el dispositivo está diseñado para resistir ataques a la disponibilidad que tendrían un impacto significativo en un individuo u organización o afectarían a muchas personas. Por ejemplo, al limitar las operaciones de una infraestructura a la que está conectada.
- **Clase 3:** además de la clase 2, el dispositivo está diseñado para proteger datos confidenciales, incluidos datos personales confidenciales.

- **Clase 4:** además de la clase 3, donde el compromiso de los datos generados o la pérdida de control tienen el potencial de afectar la infraestructura crítica o causar lesiones personales.

Para cada clase de cumplimiento, los niveles de integridad, disponibilidad y confidencialidad se muestran en la siguiente Tabla 8.

Tabla 8. Objetivos de Seguridad de la Clase de Cumplimiento - IoT Security Compliance Framework

Clase de Cumplimiento	Objetivo de Seguridad		
	Confidencialidad	Integridad	Disponibilidad
Clase 0	Básica	Básica	Básica
Clase 1	Básica	Media	Media
Clase 2	Media	Media	Alta
Clase 3	Alta	Media	Alta
Clase 4	Alta	Alta	Alta

Fuente: Tomado de <https://bit.ly/3i7s966>

2.2.9 Metodologías Ágiles

En la década de los 90's diversas metodologías ligeras de desarrollo de software surgieron, en donde el pasar de los años las denominó metodologías ágiles que ayudan a la gestión proyectos a gran escala, sin descuidar aquellos pequeños y de mediana escala. Buscando así nuevos horizontes que trajeran consigo ventajas frente a las rígidas metodologías tradicionales.

Las metodologías ágiles permiten reducir la probabilidad de fracaso por subestimaciones de costo, tiempos y funcionalidades en los proyectos de desarrollo de software (Navarro Cadavid, Fernández Martínez, & Morales Vélez, 2013).

Además, poseen dos diferencias estratégicas que las convierten en herramientas poderosas del siglo XIX para el avance efectivo de procesos de desarrollo de software; la primera se basa en que las metodologías son adaptivas y no predictivas. La segunda es que las metodologías ágiles son orientadas a las personas y no a los procesos.

Es importante destacar las características de las metodologías ágiles, entre las que se encuentran (Trigas Gallego & Domingo Troncho, n.d.):

1. Hay menos roles
2. El cliente es parte del proyecto
3. Se basa en el control empírico, basa controlar del proyecto en con base a los resultados obtenidos y en función de esto hacer las adaptaciones de acuerdo con el "espiral de mejora continua" o conocido como el ciclo PDCA (Plan, Do, Check, Act).
4. Las fases se plantean en función de los objetivos del producto; de esta forma es más fácil realizar los cambios en el transcurso del proyecto.
5. Los procesos no necesitan tanto control
6. Todo el equipo participa activamente en todas las fases del proyecto
7. Se realizan reuniones con feedback del avance y estado del proyecto
8. Comunicación fluida entre los stakeholders

Del 11 al 13 de febrero del año 2001, diecisiete representantes de diferentes metodologías ágiles como Extreme Programming, SCRUM, DSDM, Adaptive Software

Development, Crystal, Feature-Driven Development, Pragmatic Programming y otros simpatizan, se reunieron para hablar, esquiar, relajarse e intentar encontrar un terreno común. Lo que surgió, fue el Manifiesto Ágil de "Desarrollo de Software" (Software M. p., 2001). En este documento se establecen cuatro principios básicos para el desarrollo de software, los cuales contempla:

1. Individuos e interacciones sobre procesos y herramientas
2. Software funcionando sobre documentación extensiva
3. Colaboración con el cliente sobre negociación contractual
4. Respuesta ante el cambio sobre seguir un plan

Así mismo, se establecieron 12 (doce) principios, comprendiendo:

- La mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor
- Aceptar que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- Entregar software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
- Los responsables de negocio y los desarrolladores trabajan en forma junta de forma cotidiana durante todo el proyecto.
- Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- El software funcionando es la medida principal de progreso.
- Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

En la actualidad la diversificación de metodologías ágiles es amplia, sin embargo, la facilidad de usabilidad e implementación varias en un mismo proyecto, entre las que se encuentran:

- SCRUM
- Extreme Programming XP
- Kanban

2.2.9.1 SCRUM

SCRUM es un modelo de desarrollo ágil propuesto por Ikujiro Nonaka y Hirotaka Takeuchi en Japon de los años 80's, quienes identificaron un proceso eficiente, ágil y flexible de desarrollo en las empresas de manufactura industrial dedicadas a productos

tecnológicos como Fuji Xerox, Canon, Honda, Nec, Epson, Brother, 3M y HewlettPackard, quienes estaban obteniendo mejores resultados de innovación y tiempo de salida al mercado, siguiendo una secuencia de fases en donde se contaba con equipos especializados en cada una de estas. Asemejado a la forma de trabajo en equipo en el deporte de Rugby, “scrum” es el término que define el avance en formación de los jugadores, por lo que este y varios términos de este deporte son traídos a esta metodología.

Ken Schwader en 1995 presentó en OOPSLA (conferencia anual Object-Oriented Programming, Systems, Languages & Applications) una metodología de desarrollo de software basada en un ambiente scrum, usando ese mismo término, empezó a aplicarse también en la industria del software.

Modelándose y evolucionando con el pasar de los años, aplicando la filosofía de SCRUM pragmática llevada a la aplicación de valores y aspectos que permiten la ejecución de proyectos sin importar su tamaño, adaptado de forma adecuada al mundo actual en donde se presentan premisas que deben ser abordadas (Manager, 2015), proporcionando características como:

- **Autoorganización:** La gestión predictiva de un equipo que se organiza de forma autónoma, en donde no necesitan roles para la gestión de sus actividades se caracteriza por ser autónoma, permitir una autosuperación y un auto-enriquecimiento.
- **Incertidumbre:** Tomando como referente la sucesión de Fibonacci, el hecho que aumente secuencialmente de forma dinámica el tamaño de las tareas, también aumenta la incertidumbre y el margen de error, generando una “tensión” adecuada que proporciona al equipo motivación.
- **Control Moderado:** Basado en fomentar un “autocontrol entre iguales” de los miembros del equipo para no impedir la creatividad y espontaneidad.
- **Transmisión de Conocimiento:** La participación en diferentes proyectos de la organización y al compartir experiencias entre miembros del equipo transmite conocimiento.

2.2.9.1.1 Roles

Divididos en dos grupos, el primero se compone de las personas que están comprometidas con el proyecto y el marco de trabajo SCRUM (Ken Schwaber, 2016), como:

- **Propietario del Producto/Product Owner:** Es una única persona por proyecto que simplifica la comunicación con el equipo interno de la organización y toma las decisiones del cliente. Tiene el conocimiento completo del negocio, desarrollando y administrando el *Product Backlog*, y cuál es la prioridad de las funciones.
- **Equipo de Desarrollo/Development Team:** Es un equipo multifuncional de entre 3 y 9 personas (recomendado), todos aportan y colaboran con el *Product Owner* en el desarrollo del *Product Backlog*, participan en la toma de decisiones, respetan las opiniones y aportes de los demás, trabajando de forma solitaria con responsabilidades compartidas por el objetivo de cada *Sprint* y el proyecto en general.
- **Scrum Master:** Proporciona asesoría y formación para trabajar de forma autoorganizada y con responsabilidad, encargado del cumplimiento de la

metodología, asegurando que se entienda en la organización y fluya de forma adecuada.

El segundo grupo, a pesar que no son parte del proceso SCRUM, los *stakeholders* son quienes realizan retroalimentaciones y permiten determinar actividades en cada sprint, incluyendo gerentes, clientes y usuarios finales.

2.2.9.1.2 Artefactos

Los artefactos maximizan la transparencia de la información clave necesaria para que todos los actores del primer grupo - descritos en el anterior apartado - tengan el mismo entendimiento del proyecto (Palacio, 2021). Entre los que se encuentran:

- **Pila del Producto/Product Backlog:** Es la lista dinámica que enumera todas las características, funcionalidades, requisitos, mejoras y correcciones que deben incorporarse al producto a través de la evolución de los sprints. El responsable de este listado es el *producto owner*, incluyendo su contenido, disponibilidad y nivel de prioridad. Los requisitos suelen denominarse «historias de usuario», que se descomponen en «tareas» de menor tamaño, normalmente de un día de trabajo como máximo.
- **Pila del Sprint/Sprint Backlog:** Es una lista propia del equipo de desarrollo que descompone en tamaños adecuados las tareas identificadas en las «historias de usuario» del *producto backlog* para monitorizar el avance diario, identificar riesgos y problemas. Hace visible por medio de tableros físicos o herramientas colaborativas como Jira, Trello, Asana, etc., todo el trabajo incremental del equipo de desarrollo para alcanzar el objetivo del Sprint.
- **Incremento:** Es la parte del producto producida en un sprint y que se encuentra terminada, probada y operativa, en condiciones de ser entregada al cliente.
- **Gráfico de Avance/Burn Down Chart:** Se actualiza diariamente por el equipo Scrum, en donde permite comprobar el ritmo de avance y determinar si la entrega estimada del sprint es adecuada o es necesario replantearla.

2.2.9.1.3 Eventos

Los eventos son destinados para crear regularidad y minimizar la necesidad de reuniones no definidas en bloques de tiempo “*time-boxes*” con una duración máxima, entre los eventos utilizados en el marco de trabajo SCRUM se encuentran (Kniberg, 2007):

- **Sprint:** También conocido como «iteración», es el núcleo fundamental de SCRUM y es el nombre que recibe al bloque de tiempo - puede ser de una hasta seis semanas, aunque se recomienda que no exceda de un mes – en el que se crea un incremento de producto «terminado» funcional y desplegable. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint anterior.

Los Sprints contienen y consisten en la Planificación del Sprint (Sprint Planning), los Scrums Diarios (Daily Scrums), el trabajo de desarrollo, la Revisión del Sprint (Sprint Review), y la Retrospectiva del Sprint (Sprint Retrospective).

- **Planificación del Sprint/Sprint Planning:** Es la reunión que marca el inicio de cada sprint y donde participa todo el equipo SCRUM completo, teniendo un

máximo de duración de ocho horas para un Sprint de un mes, aunque este tiempo puede variar. En esta reunión se toman las prioridades y necesidades del negocio del cliente se establece cuáles y cómo van a ser las funcionalidades que se incorporarán al producto.

- **Scrum Diario/Daily Scrum:** Reunión breve de no más de 15 minutos en donde el equipo de desarrolladores sincroniza el trabajo y establece el plan para las 24 horas siguientes. Se actualiza el estado del *burn-down*.
- **Revisión del Sprint/Sprint Review:** Es la reunión que se hace al final de cada *sprint*. Habitualmente tiene una duración de una a dos horas, y tiene como objetivo ver y probar el incremento, el *product owner* y el equipo en general obtienen feedback relevante para revisar la *product backlog*.
- **Retrospectiva:** Reunión que se realiza tras la revisión de cada sprint con una duración de alrededor de tres horas. En ella el equipo realiza autoanálisis de su sobre su forma de trabajar, e identifica fortalezas y puntos débiles.

2.2.10 Interfaces de Programación de Aplicaciones

Las interfaces de programación de aplicaciones conocidas como API, son un conjunto de reglas que definen cómo las aplicaciones se comunican entre sí. Ubicadas entre una aplicación y el servidor web, actuando como una capa intermedia que procesa la transferencia de datos de forma fácil, segura y estructurada entre sistemas, que comúnmente son usadas para la comunicación efectiva entre el Backend y Frontend de una aplicación web, simplificando el desarrollo de software y la integración con librerías, brindando un nivel de flexibilidad al momento de integración (Education, 2020).

2.2.10.1 API REST

Definido por primera vez en el año 2000 por el científico informático Dr. Roy Fielding en su tesis doctoral, expone la comunicación por medio de solicitudes HTTP para realizar acciones propias de CRUD sobre una base de datos como crear, leer, actualizar y eliminar registros. Por ejemplo, haciendo uso del formato JSON una API REST usaría una solicitud GET para recuperar un registro, una solicitud POST para crear uno, una solicitud PUT para actualizar un registro y una solicitud DELETE para eliminar uno (Education, IBM, 2021).

3 INGENIERÍA DEL PROYECTO

3.1 ASPECTOS METODOLÓGICOS DE LA INVESTIGACIÓN

3.1.1 Tipo de Investigación

Dado el tipo de proyecto de investigación que se contempla en este documento, la investigación aplicada es la que se desarrolla a lo largo de este apartado. Enfocada en resolver un problema práctico por medio de la recolección, procesamiento e interpretación de información en un entorno controlado, permitiendo identificar amenazas expuestas en dispositivos IoT, de esta manera ayudar al usuario del prototipo de software a la toma de decisiones, mejorando en temas de ciberseguridad entornos empresariales y domésticos.

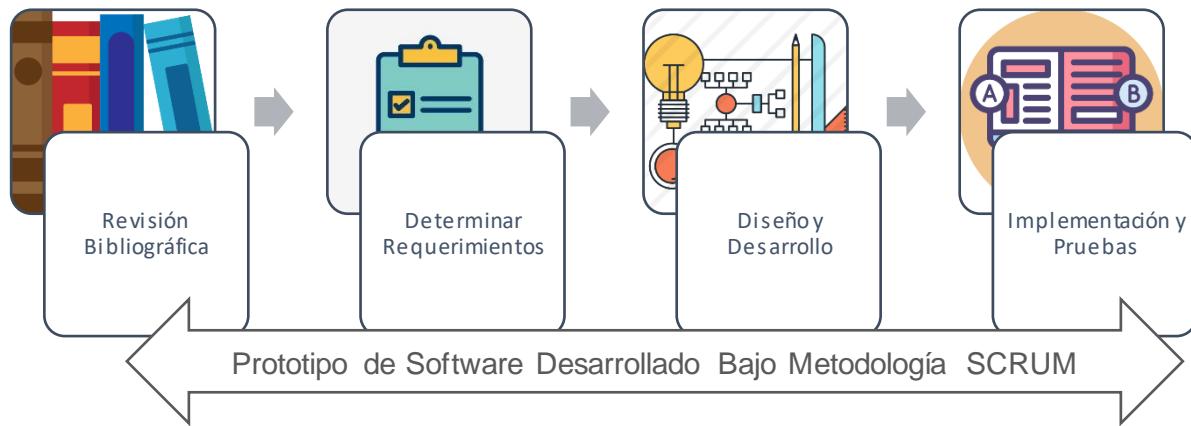
3.1.2 Metodología

La metodología seleccionada para la ejecución de este proyecto desde el punto de vista ingenieril para el proceso ordenado de codificación del código es SCRUM. Como ya se comentaba con anterioridad SCRUM, es proceso ágil ampliamente puesto en marcha en proyectos sin importar su tamaño o dimensión, lo cual se aplica de forma adecuada al desarrollo del software, que al trabajar de forma colaborativa permite obtener resultados incrementales y de calidad al producto final.

Ahora bien, sin descuidar aspectos secuenciales que permiten entender el desarrollo completo del proyecto, se realiza una revisión y recolección de bibliografías, pretendiendo acceder al conocimiento acumulado sobre ciberseguridad sobre dispositivos IoT, software preexistente para el descubrimiento de vulnerabilidades sobre estos dispositivos IoT y entender el contexto actual de esta tecnología, que ayudan a determinar los requerimientos para la codificación adecuada del software como se muestra en la Ilustración 24.

Contado con la premisa de la pila de producto, se procede con el proceso iterativo de sprints, creación de mockups, uso del lenguaje unificado de modelado (UML) y artefactos propios de SCRUM, atendiendo a las necesidades finales del producto, que ayudaran a una implementación en limpio sobre la Raspberry Pi, para hacer pruebas que confirmen los objetivos finales de este proyecto.

Ilustración 24. Metodología Desarrollo de Proyecto



Fuente: Elaboración Propia

3.1.2.1 Análisis de necesidades

La finalidad de esta etapa consiste en determinar el contexto en el cual se va a desarrollar el software e identificar requerimientos funcionales que deben ser aplicados de forma obligatoria en el prototipo de software, como herramienta colaborativa y aplicación correcta de la metodología ágil SCRUM.

3.1.2.2 Población objetivo

La población objetivo por la que se enmarca este proyecto son administradores de red en organizaciones de cualquier índole que hagan uso de dispositivos IoT en su entorno informático y se encuentran preocupados sobre la superficie de ataque a la que se encuentran expuestos de forma interna y externa. Así mismo, investigadores de ciberseguridad que se encuentren interesados en entender más acerca de la tecnología IoT y su proceso de escaneo de vulnerabilidades.

3.1.2.3 Descripción del sistema actual

En la actualidad se hacen uso de diversos programas como Nessus, OpenVAS, Qualys, etc., destinados al escaneo de vulnerabilidades sobre sistemas informáticos con equipos como computadores de escritorio y servidores, sin embargo los dispositivos IoT cuentan con una arquitectura de software completamente diferente, convirtiéndolo en una tecnología por explorar en temas de ciberseguridad y aun sin contar con un software especial que permita descubrir de forma acertada las vulnerabilidades a los que se encuentran expuestos.

El constante uso y adaptación al diario vivir de las personas sobre esta tecnología, hace más expuestos los datos personales y confidenciales en entornos empresariales, convirtiéndose en un vector de ataque poco atendido. Algunas políticas empresariales como BYOD (Bring your own device) adoptadas para facilitar a los empleados el acceso

a recursos internos como correos informáticos, bases de datos y aplicaciones, con el objetivo de facilitar la colaboración mutua entre equipos por medio de la flexibilidad, comprometen la seguridad y privacidad, y el equipo de tecnología no sabe como abordar de forma adecuada estos factores vulnerables.

3.2 DISEÑO Y DESARROLLO

Contando con la premisa que el presente proyecto se desarrolló bajo la metodología ágil SCRUM, a continuación, se muestran los artefactos que permiten entender el proceso ingenieril y nivel técnico del prototipo desarrollado.

3.2.1 Preparación del Proyecto

Conocido como el Sprint 0, permite comprender en un aspecto general la planeación requerida para la ejecución adecuada del proyecto contando con los elementos que se comentan a continuación.

3.2.1.1 Definición del proyecto

El proyecto se define bajo el desarrollo de un prototipo de software funcional para la detección de vulnerabilidades en objetos IoT que será implementado en una Raspberry Pi, contando con la premisa de su desarrollo bajo el lenguaje backend Python3 y lenguajes Fronted HTML, CSS y JavaScript, con apoyo de la biblioteca multiplataforma jQuery y la técnica de desarrollo web AJAX.

3.2.1.2 Plan de Trabajo

La planeación requerida para el proceso se enmarca en la Tabla 9.

Tabla 9. Plan de Trabajo

Artefacto	Descripción	Fecha Inicio	Fecha Fin
Product Backlog	Levantamiento de Información	24/03/2020	20/04/2020
Product Backlog	Comprender y refinar historias de usuario	23/04/2020	28/04/2020
Product Backlog	Descomposición de historias de usuario en tareas	05/05/2020	08/05/2020
Product Backlog	Priorización de tareas	11/05/2020	12/05/2020
Product Backlog	Estimación de tiempos para sprints	13/05/2020	20/05/2020

Fuente: Elaboración Propia

3.2.2 Product Backlog

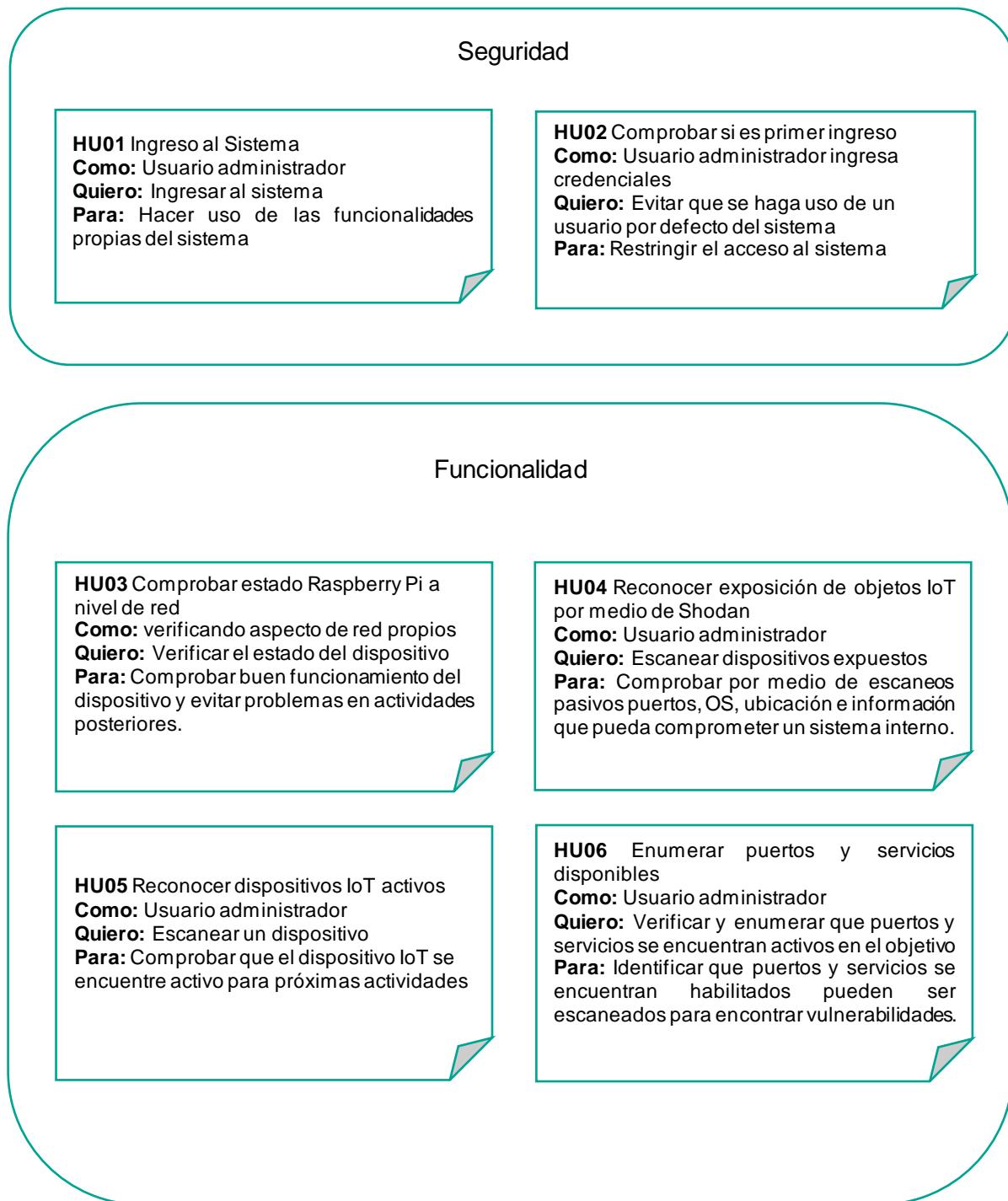
En este primer artefacto se realiza el proceso de recolección de información por medio de historias de usuario para entender cuales son los requerimiento, especificaciones y características que ayudaran a crear la lista de dinámica de la pila de

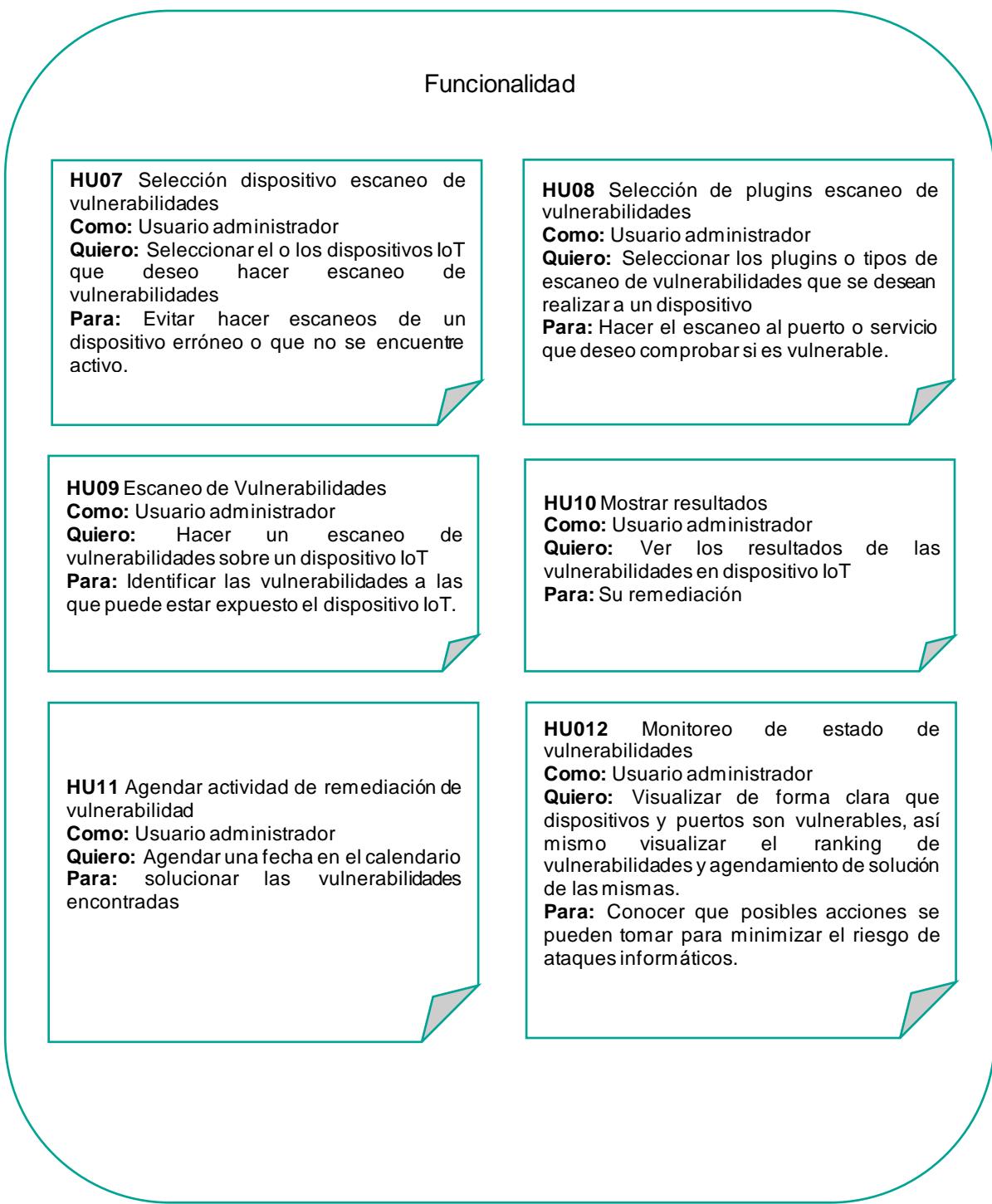
producto, haciendo estimaciones de tiempo y priorizando aquellas actividades fundamentales.

3.2.2.1 Levantamiento de Información

Primeras reuniones del product owner – correspondiente al autor del presente documento - y el stakeholder, que será asumido por el director y autor del presente proyecto. Con el objetivo de definir aspectos básicos con los que debe contar el software.

3.2.2.2 Historias de Usuario





3.2.2.3 Criterios de Aceptación Historias de Usuario

Ingreso al sistema

Como se puede observar en la Tabla 10.

Tabla 10. Ingreso a Aplicación Web

Enunciado de HU					Criterios de Aceptación			
Identificador de Historia de Usuario	Rol	Característica/Funcionalidad	Razón / Resultado	Número de Escenario	Criterio de Aceptación	Contexto	Evento	Resultado / Comportamiento Esperado
HU01	Administrador	Necesito Ingreso al sistema	Para hacer uso de las funcionalidades propias del mismo	1	Ingreso exitoso	En caso que el usuario se encuentre registrado	Cuando el ingreso de correo y credenciales sean correctos	El sistema permite el acceso a recursos propios
					No acceso al sistema	En casi que el usuario se encuentre registrado	Cuando el ingreso de correo y credenciales sean incorrectos	El sistema NO permite el ingreso a recursos internos y el sistema muestra el mensaje "Credenciales incorrectas"
					Validación ingreso	Usuario no registrado	Cuando el usuario ingresa las credenciales y oprime el botón "Entrar"	El sistema NO permite el ingreso a recursos internos y el sistema muestra el mensaje "Credenciales incorrectas"
					Campos obligatorios	Formulario de ingreso con campo de correo y contraseña obligatorios	Cuando el usuario ingresa las credenciales y oprime el botón "Entrar"	El sistema NO permite el ingreso a recursos internos y el sistema muestra en color rojo los espacios de correo o contraseña que faltan digitar

Fuente: Elaboración Propia

Comprobar si es primer ingreso

Como se puede observar en la Tabla 11.

Tabla 11. Comprobar Primer Ingreso a Sistema

Enunciado de HU					Criterios de Aceptación			
Identificador de Historia de Usuario	Rol	Característica/Funcionalidad	Razón / Resultado	Número de Escenario	Criterio de Aceptación	Contexto	Evento	Resultado / Comportamiento Esperado
HU02	Administrador	Necesito comprobar si es primer ingreso	Para evitar hacer uso de credenciales de usuario por defecto en el software	1	Ingreso exitoso	En caso que el usuario se encuentre registrado	Cuando el ingreso de correo y credenciales sean correctos, de re-dirigir a un formulario para el cambio de credenciales	El sistema a nivel de base datos realiza la actualización del parámetro "password" para el actual usuario
					No acceso al sistema	En caso que el usuario no se encuentre registrado	Cuando el ingreso de correo y credenciales sean incorrectos	El sistema NO permite el ingreso a recursos internos y el sistema muestra el mensaje "Credenciales incorrectas"
					Campos obligatorios	Formulario para cambio de credenciales por defecto del sistema	Cuando el usuario ingresa las credenciales y oprime el botón "Entrar"	El sistema NO permite el ingreso a recursos internos sin haber realizado el cambio a credenciales por defecto e ingresado un correo

Fuente: Elaboración Propia

Comprobar estado Raspberry Pi a nivel de red

Como se puede observar en la Tabla 12.

Tabla 12. Comprobar estado de Raspberry Pi

Enunciado de HU					Criterios de Aceptación			
Identificador de Historia de Usuario	Rol	Característica/Funcionalidad	Razón / Resultado	Número de Escenario	Criterio de Aceptación	Contexto	Evento	Resultado / Comportamiento Esperado
HU03	Administrador	Necesito comprobar el estado de la tarjeta Raspberry Pi a nivel de red	Para poder corroborar antes de iniciar cualquier escaneo el estado de la tarjeta Raspberry Pi	1	Estado actual	Quiero ver el estado actual a nivel de red de la tarjeta Raspberry Pi	Cuando el usuario ingrese a un modulo especialmente diseñado	Entonces el sistema trae cuales son las configuraciones actuales a nivel de red la tarjeta Raspberry Pi y lo guarda a nivel de base de datos
					Historico de estados	Cuando el usuario desee saber en una fecha determinada el estado a nivel de red de la tarjeta Raspberry Pi	Cuando se ingrese a un modulo especialmente diseñado	Entonces el sistema recupera la información guardada en la base de datos de

Fuente: Elaboración Propia

Reconocer exposición de objetos IoT por medio de Shodan

Como se puede observar en la Tabla 13.

Tabla 13. Reconocer Exposición de Objetos IoT en Internet

Enunciado de HU					Criterios de Aceptación			
Identificador de Historia de Usuario	Rol	Característica/Funcionalidad	Razón / Resultado	Número de Escenario	Criterio de Aceptación	Contexto	Evento	Resultado / Comportamiento Esperado
HU04	Administrador	Necesito identificar dispositivos IoT expuesto en internet	Para comprobar la superficie de ataque externa en mi organización	1	Busqueda de parámetro específico	Quiero por medio de un parámetro propio de mi red - puerto, OS, nombre del dispositivo, ubicación - identificar si se encuentra expuesto en internet	Cuando se ingresa en el buscador una palabra	Entonces el sistema comprueba por medio de la API de Shodan si hay algún dispositivo que concuerde con el parámetro

Fuente: Elaboración Propia

Reconocer dispositivos IoT activos

Como se puede observar en la Tabla 14.

Tabla 14. Reconocer Dispositivos IoT Activos

Enunciado de HU					Criterios de Aceptación			
Identificador de Historia de Usuario	Rol	Característica/Funcionalidad	Razón / Resultado	Número de Escenario	Criterio de Aceptación	Contexto	Evento	Resultado / Comportamiento Esperado
HU05	Administrador	Necesito poder identificar dispositivos IoT activos	Con la finalidad de tener un inventario actualizado de los recursos disponibles en mi red	1	Dispositivos activos	Quiero escanear uno o varios objetos IoT presentes en mi red	Cuando se ingresa y se guarda el formato adecuado de la dirección IPv4 o segmento de red	Entonces el sistema comprueba por medio de protocolos de red el estado de un dispositivo IoT
					Parametro incorrecto	En caso que el usuario ingrese de forma erronea la dirección IPv4 o un segmento de red	Cuando se ingresa y se guarda un escaneo	Entonces el sistema comprueba el formato de la dirección IPv4 y no arroja ningún resultado sobre el escaneo de un objeto IoT

Fuente: Elaboración Propia

Enumarar puertos y servicios disponibles

Como se puede observar en la Tabla 15.

Tabla 15. Enumerar Puertos y Servicios en Objeto IoT

Enunciado de HU					Criterios de Aceptación			
Identificador de Historia de Usuario	Rol	Característica/Funcionalidad	Razón / Resultado	Número de Escenario	Criterio de Aceptación	Contexto	Evento	Resultado / Comportamiento Esperado
HU06	Administrador	Necesito poder identificar los puertos y servicios disponibles en objetos IoT	Con la finalidad de conocer las versiones y nivel de exposición	1	Estado de puertos y servicios	Quiero escanear uno o varios objetos IoT presentes en mi red	Cuando se ingresa y se guarda de formato adecuado la dirección IPv4 o segmento de red	Entonces el sistema comprueba por medio de protocolos de red el estado de servicios y puertos comunes en objetos IoT
					Parametro incorrecto	En caso que el usuario ingrese de forma erronea la dirección IPv4 o un segmento de red	Cuando se ingresa y se guarda un escaneo	Entonces el sistema comprueba el formato de la dirección IPv4 y no arroja ningún resultado sobre el escaneo de puertos y servicios de un objeto IoT

Fuente: Elaboración Propia

Selección dispositivo escaneo de vulnerabilidades

Como se puede observar en la Tabla 16.

Tabla 16. Seleccionar Dispositivo IoT para Escaneo de Vulnerabilidades

Enunciado de HU					Criterios de Aceptación			
Identificador de Historia de Usuario	Rol	Característica/Funcionalidad	Razón / Resultado	Número de Escenario	Criterio de Aceptación	Contexto	Evento	Resultado / Comportamiento Esperado
HU07	Administrador	Necesito seleccionar los dispositivos IoT activos de mi preferencia para hacer el escaneo de vulnerabilidades	Para evitar consumo de recursos innecesarios o escanear algún dispositivo IoT critico en una organización	1	Selección de objeto IoT	Quiero poder seleccionar los dispositivos IoT activos de mi preferencia	Cuando los seleccione y los guarde	Entonces el sistema los mantiene en la base de datos para la posterior ejecución del escaneo de vulnerabilidades

Fuente: Elaboración Propia

Selección de plugins para escaneo de vulnerabilidades

Como se puede observar en la Tabla 17.

Tabla 17. Selección de Plugins para Escaneo de Vulnerabilidades

Enunciado de HU					Criterios de Aceptación			
Identificador de Historia de Usuario	Rol	Característica/Funcionalidad	Razón / Resultado	Número de Escenario	Criterio de Aceptación	Contexto	Evento	Resultado / Comportamiento Esperado
HU08	Administrador	Necesito seleccionar los plugins	Para hacer el escaneo de vulnerabilidades sobre un puerto o servicio en específico	1	Selección de plugins	Quiero poder seleccionar los plugins de mi preferencia	Cuando los seleccione y los guarde	Entonces el sistema los mantiene en la base de datos para la posterior ejecución del escaneo de vulnerabilidades

Fuente: Elaboración Propia

Iniciar escaneo de vulnerabilidades

Como se puede observar en la Tabla 18.

Tabla 18. Iniciar Escaneo de Vulnerabilidades Sobre Objeto IoT

Enunciado de HU					Criterios de Aceptación			
Identificador de Historia de Usuario	Rol	Característica/Funcionalidad	Razón / Resultado	Número de Escenario	Criterio de Aceptación	Contexto	Evento	Resultado / Comportamiento Esperado
HU09	Administrador	Necesito iniciar el escaneo de vulnerabilidades	Para descubrir las amenazas a las que se encuentra expuesto un dispositivo IoT	1	Sistema de detección de vulnerabilidades	Quiero activar el mecanismo para la detección de vulnerabilidades sobre un objeto IoT	Cuando se presione el botón "Iniciar Escaneo"	Entonces el sistema toma los objetos IoT y plugins previamente seleccionados y evalua de forma intrusiva que servicios pueden ser explotados, mediante el uso de diversas técnicas

Fuente: Elaboración Propia

Mostrar resultado de escaneo de vulnerabilidades

Como se puede observar en la Tabla 19.

Tabla 19. Mostrar Resultado Escaneo de Vulnerabilidades de Objeto IoT

Identificador de Historia de Usuario	Rol	Característica/Funcionalidad	Razón / Resultado	Número de Escenario	Criterios de Aceptación				Resultado / Comportamiento Esperado
					Criterio de Aceptación	Contexto	Evento		
HU10	Administrador	Necesito mostrar los resultados de un escaneo de vulnerabilidades	Para identificar aquellas vulnerabilidades que ponen en peligro una red	1	Vulnerabilidad igual a 0.0	Al detectarse una vulnerabilidad categorizada igual 0.0	Asignar clasificación de vulnerabilidad en "Info", correspondiente a información propia del sistema que no compromete ningún activo o servicio	Entonces el sistema le asigna un color "Azul"	
				2	Vulnerabilidad entre 0.1 - 3.9	Al detectarse una vulnerabilidad categorizada entre 0.1 - 3.9	Asignar clasificación de vulnerabilidad en "Bajo", correspondiente a vulnerabilidades generalmente que requiere acceso al sistema local o físico. No tienen impacto en el negocio	Entonces el sistema le asigna un color "Verde"	
				3	Vulnerabilidad entre 4.0 - 6.9	Al detectarse una vulnerabilidad categorizada entre 4.0 - 6.9	Asignar clasificación de vulnerabilidad en "Medio", correspondiente a vulnerabilidades de DoS, acceso limitado a recursos o requieren privilegios de usuario para una explotación exitosa	Entonces el sistema le asigna un color "Amarillo"	
				4	Vulnerabilidad entre 7.0 - 8.9	Al detectarse una vulnerabilidad categorizada entre 7.0 - 8.9	Asignar clasificación de vulnerabilidad en "Alto", correspondiente a vulnerabilidades de pérdida significativa de datos o tiempo de inactividad. Así mismo, acceso a privilegios elevados de usuario	Entonces el sistema le asigna un color "Naranja"	
				5	Vulnerabilidad entre 9.0 - 10.0	Al detectarse una vulnerabilidad categorizada entre 9.0 - 10.0	Asignar clasificación de vulnerabilidad en "Crítico", correspondiente a una vulnerabilidad que resulte en un compromiso a nivel de raíz de los servidores o dispositivos de infraestructura, así mismo explotación sencilla	Entonces el sistema le asigna un color "Roja"	
				6	Aspecto de visualización de información	Quiero poder ver de forma organizada y clara el resultado de los escaneos de vulnerabilidades	Cuando ingrese a un escaneo en específico	Entonces el sistema por medio de gráficas, tablas y demás muestra la información	

Fuente: Elaboración Propia

Agendar actividad de remediación de vulnerabilidad

Como se puede observar en la Tabla 20.

Tabla 20. Agendar Actividad de Remediación de Vulnerabilidad

Identificador de Historia de Usuario	Rol	Característica/Funcionalidad	Razón / Resultado	Número de Escenario	Criterios de Aceptación				Resultado / Comportamiento Esperado
					Criterio de Aceptación	Contexto	Evento		
HU11	Administrador	Necesito agendar una fecha/hora	Para poder cerrar vulnerabilidades en dispositivos IoT que puedan comprometer una red	1	Agenda actividad de remediación	Quiero poder seleccionar en un calendario la fecha/hora	Cuando presione en rango de tiempo específico	Entonces el sistema trae las vulnerabilidades latentes en un dispositivo IoT y guarda a nivel de base de datos la información	
				2	Especificar estado de actividad de remediación	Quiero poder conocer en que estado se encuentra una actividad de remediación de una vulnerabilidad	Cuando presione una actividad de remediación previamente creada	Entonces el sistema muestra las actualizaciones realizadas por el equipo de TI o grupo designado	

Fuente: Elaboración Propia

Monitoreo de estado de vulnerabilidades

Como se puede observar en la Tabla 21.

Tabla 21. Monitorear de Manera Global el Estado de una Red

Enunciado de HU					Criterios de Aceptación			
Identificador de Historia de Usuario	Rol	Característica/Funcionalidad	Razón / Resultado	Número de Escenario	Criterio de Aceptación	Contexto	Evento	Resultado / Comportamiento Esperado
HU12	Administrador	Necesito monitorear de manera global el estado de una red	Para poder entender de forma resumida el estado de amenaza latente	1	Resumen global de amenazas	Quiero ver en diversas gráficas dispositivo IoT, puerto, vulnerabilidades y actividades de remediación que permitan entender la superficie de ataque en la que se pueda encontrar una red	Cuando se ingrese a la pantalla principal	Entonces el sistema de manera automática recupera la información guardada en base de datos de escaneo de vulnerabilidades y los grafica

Fuente: Elaboración Propia

3.2.2.4 Product Backlog

Teniendo presentes las historias de usuario, se procede con la construcción de la lista de requerimientos priorizados que se harán uso a lo largo del desarrollo de la aplicación web bajo el marco de trabajo SCRUM, que se encuentra sujeta a posibles cambios, con el objetivo de hacer entrega de elementos funcionales acorde a lo necesitado. El producto backlog puede observarse en la Tabla 22.

Tabla 22. Product Backlog

Prioridad	Ítem	Descripción	Estimación en Horas	Por
Muy alta				
	1	Interfaz gráfica del usuario	378	LF
		Base de datos		
	2	Diseño y creación de base de datos	24	LF
	3	Comunicación con base de datos	6	LF
	4	Conexión segura a base de datos	2	LF
	5	Credenciales seguras en base de datos	8	LF
		Autenticación y autorización		
	6	Cambio de credenciales genéricas	12	LF
	7	El sistema controlará el acceso a escaneos propios y lo permitirá solamente a usuarios autorizados	8	LF
		Vulnerabilidades		
	8	Detección de Vulnerabilidades	16	
	9	Identificar y clasificar la criticidad de vulnerabilidades	11	LF
	10	Clasificar remediación de vulnerabilidades	8	LF
	11	Evaluuar a nivel de seguridad el impacto para dispositivo IoT	9	LF
	12	Por cada dispositivo IoT Identificar y clasificar las vulnerabilidades encontradas	12	LF
	13	Evaluuar nivel de impacto a nivel de seguridad para cada puerto descubierto	6	LF
Alta				
		Autenticación y autorización		
	14	Validación de correo	4	LF

	15	Validación de contraseña	4	LF
	Modulo gestión de IoT			
	16	Crear escaneo	16	LF
	17	Modificar escaneo	12	LF
	18	Eliminar escaneo	10	LF
	19	Resultado escaneo	22	LF
	20	Organizar y clasificar en cada campo el parámetro encontrado en los dispositivos IoT	11	LF
	21	Barra de progreso escaneo	5	LF
	22	Iniciar escaneo	23	LF
	Modulo escaneo de vulnerabilidades			
	23	Seleccionar Objetivo	6	LF
	24	Seleccionar Plugins	42	LF
	25	Iniciar escaneo vulnerabilidades	12	LF
	26	Barra de progreso escaneo vulnerabilidades	8	LF
	27	Resultado escaneo vulnerabilidades	6	LF
	28	Lista de hallazgos escaneo de vulnerabilidades	13	LF
	29	Mostrar resultado de explotación	3	LF
	Modulo gestión de vulnerabilidades			
	30	Calendario principal	8	LF
	31	Crear gestión de vulnerabilidad	19	LF
	32	Modificar gestión de vulnerabilidades	11	LF
	33	Borrar gestión de vulnerabilidades	9	LF
	34	Mostrar vulnerabilidades detectadas para su gestión	11	LF
Media				
		Modulo dashboard		
	35	Gráfica histórico avance de vulnerabilidades	13	LF
	36	Gráfica resumen global de vulnerabilidades	13	LF
	37	Gráfica resumen global de dispositivos	11	LF
	38	Gráfica resumen global de puertos	16	LF
	39	Gráfica resumen gestión de vulnerabilidades	14	LF
	40	Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.	1	LF
	Modulo estado de dispositivo			
	41	Verificar estado actual dispositivo	9	LF
	42	Histórico de estados de dispositivo	8	LF
	Modulo Shodan			
	43	Integración API Shodan	3	LF
	44	Buscador Shodan	11	LF
	Modulo escaneo de vulnerabilidades			
	45	Gráfica resultado escaneo de vulnerabilidades	11	LF
	46	Apartado resumen de escaneo vulnerabilidades	3	LF
	47	Expandir/Contraer ítems resultado escaneo vulnerabilidades	3	LF

	Base de datos		
48	La base de datos será implementada con trazas de auditoría	4	LF
	Usabilidad y rendimiento		
49	El sistema debe ser capaz de procesar 3 tareas simultaneas	4	LF
50	El sistema debe proporcionar mensajes de error/satisfactorios que sean informativos y orientados a usuario final	11	LF
Baja			
	Modulo gestión de IoT		
51	Tarjetas estado escaneos	10	LF
52	Copiar tabla escaneo	7	LF
53	Filtro escaneo	6	LF
54	Exportar tabla escaneos	3	LF
55	Exportar resultado escaneo	3	LF
56	Buscar dispositivo en resultado de escaneo	7	LF
	Modulo escaneo de vulnerabilidades		
57	Tarjetas estado escaneos de vulnerabilidades	7	LF
58	Copiar tabla escaneo vulnerabilidades	2	LF
59	Filtro escaneo vulnerabilidades	2	LF
60	Exportar tabla escaneos vulnerabilidades	5	LF
61	Copiar tabla resultado escaneo de vulnerabilidades	6	LF
62	Filtro tabla resultado escaneo de vulnerabilidades	8	LF
63	Exportar tabla resultado escaneo vulnerabilidades	3	LF
64	Buscar vulnerabilidad en resultado escaneo vulnerabilidades	5	LF
65	Buscar dispositivo en resultado de escaneo vulnerabilidades	5	LF
66	Buscar puerto en resultado de escaneo vulnerabilidades	5	LF
	Modulo gestión de vulnerabilidades		
67	Filtros de tiempo calendario	6	LF
Muy Baja			
	Usabilidad y rendimiento		
68	Mostrar correo de usuario logueado	6	LF
69	Desplegar/contrae módulos	4	LF
70	Instalador	10	LF

Fuente: Elaboración Propia

3.2.3 Análisis de Requerimientos

Contando con información inicial que permite entender la dimensión del proyecto a nivel de desarrollo de software y requerimientos básicos de la solución que describen las capacidades expresadas en términos de comportamiento. A continuación, se realiza la segregación estableciendo los requerimientos funcionales y no funcionales, ayudando comprender la relevancia en la codificación del software.

3.2.3.1 Requerimientos Funcionales

Los requerimientos funcionales del proyecto se pueden apreciar en la Tabla 23, que se muestra a continuación.

Tabla 23. Requerimientos Funcionales

Identificador	Requerimiento	Caso de Uso	Actor
RF01	A nivel de aplicación web se debe contar con un proceso de autenticación y autorización recursos propios del sistema	Inicio de sesión	Administrador
RF02	El sistema debe detectar el uso de credenciales por defecto en un usuario y solicitar un cambio de las mismas por medio web	Cambio de credenciales por defecto	Administrador
RF03	El sistema debe permitir tener un monitoreo en tiempo real a nivel de red sobre la Raspberry Pi, garantizando que procesos posteriores se lleven de forma adecuada	Comprobar estado de Raspberry Pi	Administrador
RF04	El sistema debe permitir crear, modificar y eliminar a nivel de base de datos los escaneos sobre dispositivos IoT para el descubrimiento de estado, puertos y servicios.	Gestión IoT	Administrador
RF05	El sistema debe permitir por medio de la API de Shodan, poder hacer la búsqueda de posibles objetos IoT que sean accesibles por medio de Internet en una red objetivo decidida por el usuario	Consultas Shodan	Administrador
RF06	El sistema debe permitir seleccionar el objeto u objetos IoT descubiertos sobre los cuales se hará el escaneo de vulnerabilidades	Seleccionar Objeto IoT	Administrador
RF07	El sistema debe permitir seleccionar el plugin o plugins deseados para llevar a cabo el escaneo sobre un puerto o servicio deseado.	Seleccionar Plugin	Administrador
RF08	El sistema debe permitir mostrar de forma organizada en tablas y graficas el resultado de un escaneo de vulnerabilidades sobre un objeto IoT	Resultado de Escaneo de Vulnerabilidades	Administrador
RF09	Manejando el sistema de puntuación de vulnerabilidades (CVSS) dependiendo de la criticidad el resultado del escaneo varía así: 0.1 – 3.9 = Bajo 4.0 – 6.9 = Medio 7.0 – 8.9 = Alto 9.0 – 10.0 = Crítico	Puntuación de Vulnerabilidad	Administrador
RF10	La clasificación y modelo de remediación asignada a cada vulnerabilidad estará basada en Internet of Things de OWASP y el IoT Security Compliance Framework de Internet of Things Security Foundation	Modelos de Referencia Clasificación de Vulnerabilidades	Administrador
RF11	Por cada vulnerabilidad encontrada se debe dar un resumen general, explicando en que consiste la	Detalle de Vulnerabilidad	Administrador

	vulnerabilidad, así mismo un resultado de explotación de la vulnerabilidad y un proceso de recomendación que ayude a mitigar esta vulnerabilidad en el objeto IoT.		
RF12	El sistema debe permitir crear, modificar y eliminar actividades de remediación de vulnerabilidades	Gestión de Vulnerabilidades	Administrador
RF13	El sistema debe permitir controlar el estado de una remediación de vulnerabilidades por el usuario final	Administración de Remediación de Vulnerabilidades	Administrador
RF14	En la pantalla inicial del sitio web luego del ingreso, se debe mostrar un dashboard con información en gráficos relevantes de las vulnerabilidades en el tiempo, estado de un objeto IoT, puertos más vulnerables, etc., que ayuden a la toma de decisiones.	Dashboard	Administrador
RF15	El sistema debe contar con una barra de progreso en aquellos apartados en donde se hagan procesos en background que puedan tomar tiempo, ayudando al usuario a entender cuando haya terminado.	Barra de Progreso	Administrador
RF16	El sistema debe permitir conocer cuando un escaneo se inicie de forma adecuada o sea necesario algún tipo de intervención debido a la falta de un parámetro o suceso aislado que impida su correcta ejecución.	Inicio de Escaneo	Administrador

Fuente: Elaboración Propia

3.2.3.2 Requerimientos No Funcionales

Se identificaron los siguientes requerimientos no funcionales sobre el proyecto (ver Tabla 24).

Tabla 24. Requerimientos No Funcionales

Ítem	Atributo de Calidad	Descripción
RNF01	Disponibilidad	El sistema debe estar disponible en un 99,99% de las veces en que sea solicitado su acceso.
RNF02	Tolerancia a fallos	El sistema realiza un alojamiento externo - bien sea en la nube o medio extraíble - de la base de datos para evitar perdida de información.
RNF03	Seguridad	Las comunicaciones web realizadas entre cliente-servidor deben estar encriptadas por medio de criptográfica asimétrica, evitando envío de datos en texto plano.
RNF04	Usabilidad	El sistema guarda a nivel de base de datos escaneos que hayan sido eliminados para su posterior recuperación o restablecimiento.
RNF05	Compatibilidad	A nivel de aplicación web debe ser compatible con la mayoría de los navegadores que soporte HTML5 y JavaScript, disponibles en el mercado actual.
RNF06	Portabilidad	El sistema podrá ser ejecutado de manera fluida en sistemas operativos Windows independientemente de su arquitectura o versión.

Fuente: Elaboración Propia

3.2.4 Desarrollo de Casos de Uso

A pesar que en una mitología ágil no es obligatorio utilizar los casos de uso, para este caso en particular se utilizará para explicar la interacción entre el usuario y el sistema, brindando una comprensión más adecuada al software.

3.2.4.1 Formato Caso de Uso General

En la siguiente Tabla 25 se aprecia de forma general el caso de uso que comprende a la herramienta de software.

Tabla 25. Caso de Uso General

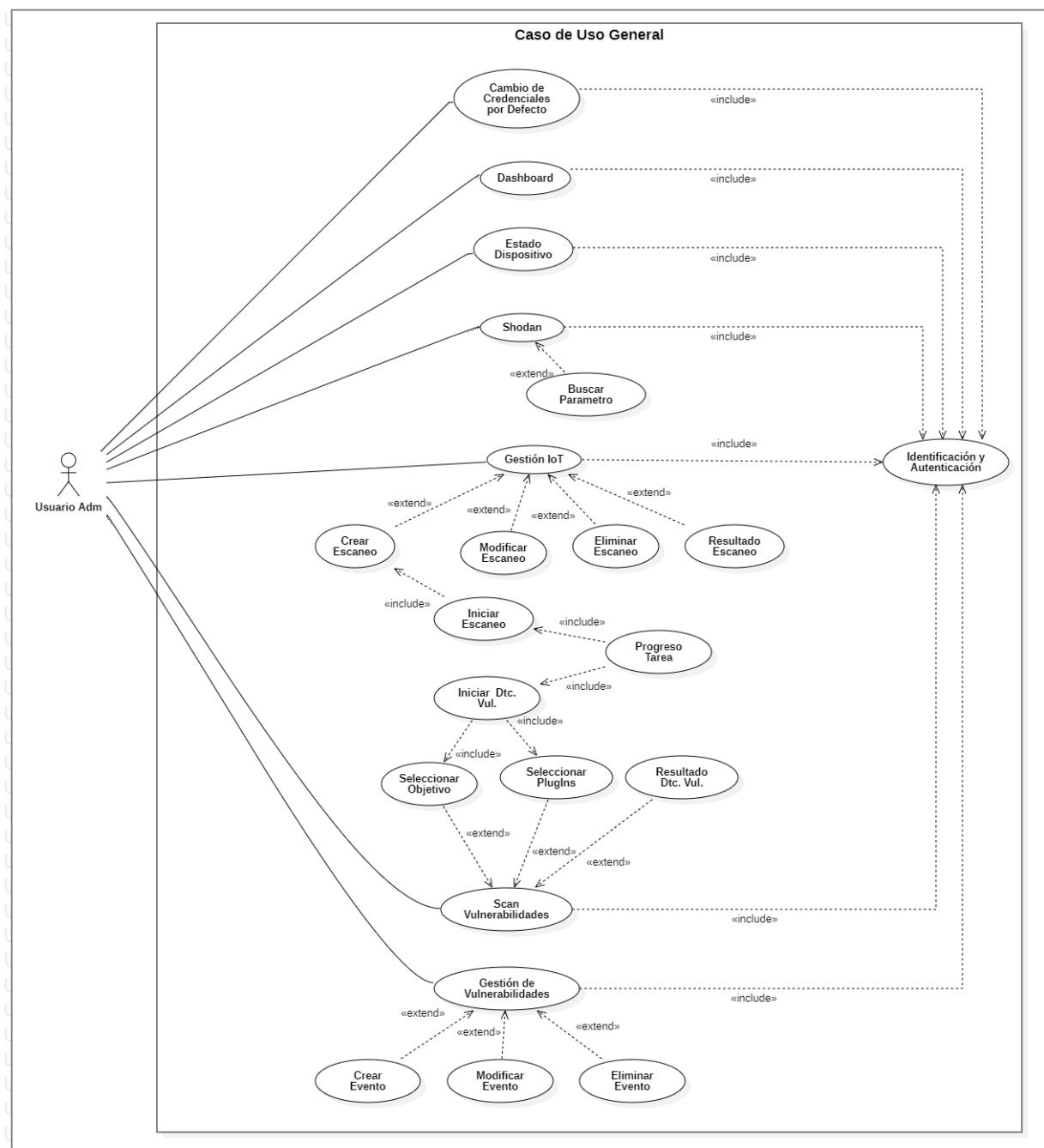
Caso de Uso	Caso de uso general																										
Actor	Administrador																										
Tipo	Primario																										
Descripción	Especificación de alto nivel de uso de sistema																										
Precondición	A nivel de base de datos debe estar registrado un usuario, bien sea genérico o personalizado para poder ingresar a la aplicación vía web. Debe estar activo en escucha el proceso que ejecuta tareas en background.																										
Flujo Básico	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El usuario abre la aplicación web desde el navegador</td></tr> <tr> <td>2</td><td>El usuario se autentica con las credenciales de asignadas e ingresa a la página inicial de Dashboard</td></tr> <tr> <td>3</td><td>El usuario verifica el estado a nivel de red de la Raspberry Pi</td></tr> <tr> <td>4</td><td>El usuario verifica los dispositivos IoT expuestos en Internet por medio del módulo de Shodan</td></tr> <tr> <td>5</td><td>El usuario crea un escaneo en el módulo de gestión IoT</td></tr> <tr> <td>6</td><td>El usuario ejecuta el escaneo activo sobre el objeto IoT</td></tr> <tr> <td>7</td><td>El usuario selecciona el objeto u objetos IoT para escaneo de vulnerabilidades</td></tr> <tr> <td>8</td><td>El usuario selecciona la serie de plugins que desee para el escaneo de vulnerabilidades</td></tr> <tr> <td>9</td><td>El usuario inicia el escaneo de vulnerabilidades sobre el objeto IoT</td></tr> <tr> <td>10</td><td>El usuario crea una actividad de remediación en el módulo de gestión de vulnerabilidades</td></tr> <tr> <td>11</td><td>El usuario hace seguimiento de actividades de remediación y estado de red a nivel general en el dashboard</td></tr> <tr> <td>12</td><td>Finaliza caso de uso</td></tr> </tbody> </table>	Paso	Acción	1	El usuario abre la aplicación web desde el navegador	2	El usuario se autentica con las credenciales de asignadas e ingresa a la página inicial de Dashboard	3	El usuario verifica el estado a nivel de red de la Raspberry Pi	4	El usuario verifica los dispositivos IoT expuestos en Internet por medio del módulo de Shodan	5	El usuario crea un escaneo en el módulo de gestión IoT	6	El usuario ejecuta el escaneo activo sobre el objeto IoT	7	El usuario selecciona el objeto u objetos IoT para escaneo de vulnerabilidades	8	El usuario selecciona la serie de plugins que desee para el escaneo de vulnerabilidades	9	El usuario inicia el escaneo de vulnerabilidades sobre el objeto IoT	10	El usuario crea una actividad de remediación en el módulo de gestión de vulnerabilidades	11	El usuario hace seguimiento de actividades de remediación y estado de red a nivel general en el dashboard	12	Finaliza caso de uso
Paso	Acción																										
1	El usuario abre la aplicación web desde el navegador																										
2	El usuario se autentica con las credenciales de asignadas e ingresa a la página inicial de Dashboard																										
3	El usuario verifica el estado a nivel de red de la Raspberry Pi																										
4	El usuario verifica los dispositivos IoT expuestos en Internet por medio del módulo de Shodan																										
5	El usuario crea un escaneo en el módulo de gestión IoT																										
6	El usuario ejecuta el escaneo activo sobre el objeto IoT																										
7	El usuario selecciona el objeto u objetos IoT para escaneo de vulnerabilidades																										
8	El usuario selecciona la serie de plugins que desee para el escaneo de vulnerabilidades																										
9	El usuario inicia el escaneo de vulnerabilidades sobre el objeto IoT																										
10	El usuario crea una actividad de remediación en el módulo de gestión de vulnerabilidades																										
11	El usuario hace seguimiento de actividades de remediación y estado de red a nivel general en el dashboard																										
12	Finaliza caso de uso																										
Flujo Alternativo	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>2.1</td><td>Si el sistema detecta el uso de credenciales por defecto, se redireccionará al formulario de cambio de credenciales</td></tr> <tr> <td>3.1</td><td>El sistema detecta algún valor a nivel de red que evita el posterior funcionamiento correcto de un escaneo</td></tr> <tr> <td>6.1</td><td>El usuario no ejecuta el escaneo activo sobre el objeto IoT</td></tr> <tr> <td>6.2</td><td>El sistema detecta un error o falta de algún parámetro que impide ejecutar el escaneo activo sobre el objeto IoT</td></tr> <tr> <td>7.1</td><td>El sistema no muestra la lista de objetos IoT debido a que no se ejecutó un escaneo activo</td></tr> <tr> <td>8.1</td><td>El sistema no muestra la lista de plugins debido a que no se seleccionó un objeto IoT para hacer el escaneo de vulnerabilidades</td></tr> <tr> <td>9.1</td><td>El usuario no ejecuta el escaneo de vulnerabilidades</td></tr> <tr> <td>9.2</td><td>El sistema detecta un error o falta de algún que impide ejecutar el escaneo de vulnerabilidades</td></tr> </tbody> </table>	Paso	Acción	2.1	Si el sistema detecta el uso de credenciales por defecto, se redireccionará al formulario de cambio de credenciales	3.1	El sistema detecta algún valor a nivel de red que evita el posterior funcionamiento correcto de un escaneo	6.1	El usuario no ejecuta el escaneo activo sobre el objeto IoT	6.2	El sistema detecta un error o falta de algún parámetro que impide ejecutar el escaneo activo sobre el objeto IoT	7.1	El sistema no muestra la lista de objetos IoT debido a que no se ejecutó un escaneo activo	8.1	El sistema no muestra la lista de plugins debido a que no se seleccionó un objeto IoT para hacer el escaneo de vulnerabilidades	9.1	El usuario no ejecuta el escaneo de vulnerabilidades	9.2	El sistema detecta un error o falta de algún que impide ejecutar el escaneo de vulnerabilidades								
Paso	Acción																										
2.1	Si el sistema detecta el uso de credenciales por defecto, se redireccionará al formulario de cambio de credenciales																										
3.1	El sistema detecta algún valor a nivel de red que evita el posterior funcionamiento correcto de un escaneo																										
6.1	El usuario no ejecuta el escaneo activo sobre el objeto IoT																										
6.2	El sistema detecta un error o falta de algún parámetro que impide ejecutar el escaneo activo sobre el objeto IoT																										
7.1	El sistema no muestra la lista de objetos IoT debido a que no se ejecutó un escaneo activo																										
8.1	El sistema no muestra la lista de plugins debido a que no se seleccionó un objeto IoT para hacer el escaneo de vulnerabilidades																										
9.1	El usuario no ejecuta el escaneo de vulnerabilidades																										
9.2	El sistema detecta un error o falta de algún que impide ejecutar el escaneo de vulnerabilidades																										

	10.1	El sistema no lista los escaneos con las vulnerabilidades encontradas que permitan crear una actividad de remediación
Postcondición	Uso general de sistema	

Fuente: Elaboración Propia

Se muestra a continuación en la Ilustración 25 el caso de uso general.

Ilustración 25. Caso de Uso General



Fuente: Elaboración Propia

3.2.4.2 Formato Caso de Uso Autenticación de Usuario

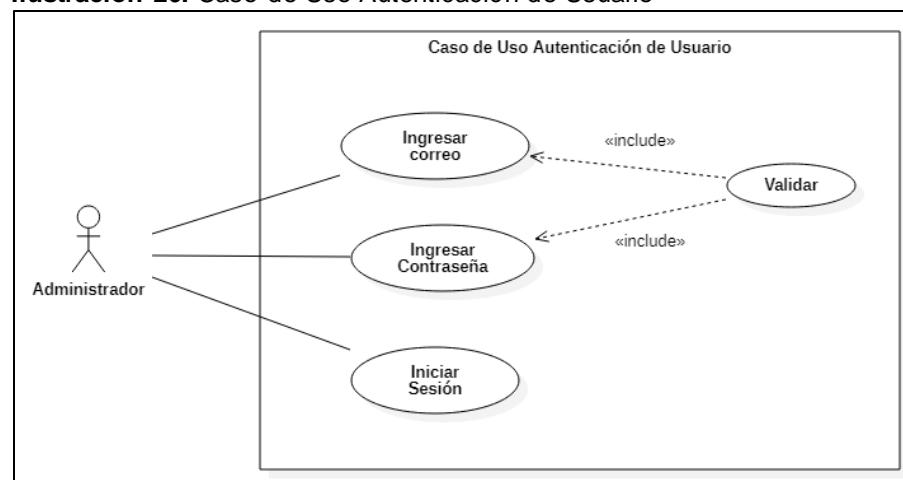
Tabla 26. Caso de Uso Autenticación de Usuario

Caso de Uso	Autenticación de Usuario														
Actor	Administrador														
Tipo	Primario														
Descripción	Realiza la validación de las credenciales del usuario para permitirle acceso a la plataforma. Todos los usuarios deberán identificarse para acceder a cualquier parte del sistema.														
Precondición	El usuario debe estar registrado														
Flujo Básico	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El administrador abre la aplicación desde el navegador</td> </tr> <tr> <td>2</td> <td>El sistema muestra el formulario de autenticación</td> </tr> <tr> <td>3</td> <td>El administrador ingresa su correo y contraseña</td> </tr> <tr> <td>4</td> <td>El sistema muestra la página principal del dashboard</td> </tr> <tr> <td>5</td> <td>El administrador cierra la sesión</td> </tr> <tr> <td>6</td> <td>Finaliza el caso de uso</td> </tr> </tbody> </table>	Paso	Acción	1	El administrador abre la aplicación desde el navegador	2	El sistema muestra el formulario de autenticación	3	El administrador ingresa su correo y contraseña	4	El sistema muestra la página principal del dashboard	5	El administrador cierra la sesión	6	Finaliza el caso de uso
Paso	Acción														
1	El administrador abre la aplicación desde el navegador														
2	El sistema muestra el formulario de autenticación														
3	El administrador ingresa su correo y contraseña														
4	El sistema muestra la página principal del dashboard														
5	El administrador cierra la sesión														
6	Finaliza el caso de uso														
Flujo Alternativo	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>3.1</td> <td>Si el sistema detecta el uso de credenciales por defecto, se redireccionará al formulario de cambio de credenciales</td> </tr> <tr> <td>3.2</td> <td>Si el correo o la contraseña son incorrectos muestra un mensaje genérico de credenciales erróneas y retorna al formulario de autenticación paso 2</td> </tr> </tbody> </table>	Paso	Acción	3.1	Si el sistema detecta el uso de credenciales por defecto, se redireccionará al formulario de cambio de credenciales	3.2	Si el correo o la contraseña son incorrectos muestra un mensaje genérico de credenciales erróneas y retorna al formulario de autenticación paso 2								
Paso	Acción														
3.1	Si el sistema detecta el uso de credenciales por defecto, se redireccionará al formulario de cambio de credenciales														
3.2	Si el correo o la contraseña son incorrectos muestra un mensaje genérico de credenciales erróneas y retorna al formulario de autenticación paso 2														
Postcondición	Autenticación exitosa de usuario														

Fuente: Elaboración Propia

En la siguiente Ilustración 26 se evidencia el caso de uso de autenticación.

Ilustración 26. Caso de Uso Autenticación de Usuario



Fuente: Elaboración Propia

3.2.4.3 Formato Caso de Uso Cambio de Credenciales por Defecto

Tabla 27. Caso de Uso Cambio de Credenciales por Defecto

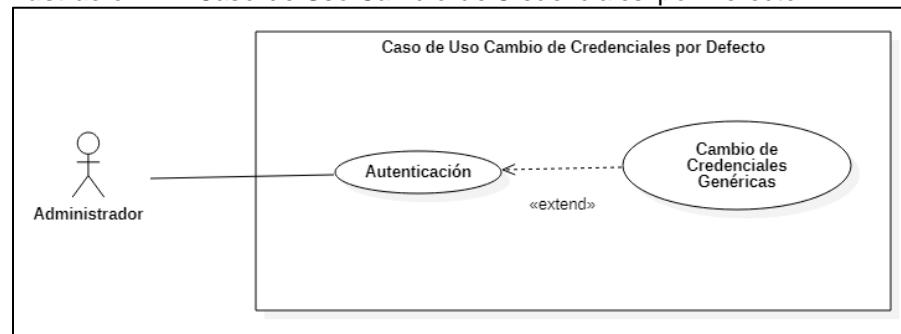
Caso de Uso	Cambio de Credenciales por Defecto
Actor	Administrador
Tipo	Primario
Descripción	Permite realizar el cambio de credenciales al usuario por defecto o posteriores usuarios creados en el sistema. Evitando acceso a información de usuarios no autorizados.

Precondición	El usuario ha ingresado correo y credenciales de forma exitosa para el usuario genérico																
Flujo Básico	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El sistema muestra el formulario de autenticación</td> </tr> <tr> <td>2</td> <td>El administrador ingresa su correo y contraseña</td> </tr> <tr> <td>3</td> <td>El sistema detecta que son credenciales genéricas</td> </tr> <tr> <td>4</td> <td>El sistema redirecciona al formulario para cambio de credenciales</td> </tr> <tr> <td>5</td> <td>El administrador confirma cambio de credenciales</td> </tr> <tr> <td>6</td> <td>El sistema redirecciona al formulario de autenticación</td> </tr> <tr> <td>7</td> <td>Finaliza el caso de uso</td> </tr> </tbody> </table>	Paso	Acción	1	El sistema muestra el formulario de autenticación	2	El administrador ingresa su correo y contraseña	3	El sistema detecta que son credenciales genéricas	4	El sistema redirecciona al formulario para cambio de credenciales	5	El administrador confirma cambio de credenciales	6	El sistema redirecciona al formulario de autenticación	7	Finaliza el caso de uso
Paso	Acción																
1	El sistema muestra el formulario de autenticación																
2	El administrador ingresa su correo y contraseña																
3	El sistema detecta que son credenciales genéricas																
4	El sistema redirecciona al formulario para cambio de credenciales																
5	El administrador confirma cambio de credenciales																
6	El sistema redirecciona al formulario de autenticación																
7	Finaliza el caso de uso																
Flujo Alternativo	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>2.1</td> <td>Si el correo o la contraseña son incorrectos muestra un mensaje genérico de credenciales erróneas y retorna al formulario de autenticación paso 1</td> </tr> <tr> <td>3.1</td> <td>Si las credenciales ingresadas no corresponden a credenciales genéricas se redirecciona al dashboard principal</td> </tr> <tr> <td>6.1</td> <td>En caso que algún parámetro del cambio de credenciales sea incorrecto se resaltará en color rojo el campo faltan o erróneo y no permitirá avanzar al paso 7</td> </tr> </tbody> </table>	Paso	Acción	2.1	Si el correo o la contraseña son incorrectos muestra un mensaje genérico de credenciales erróneas y retorna al formulario de autenticación paso 1	3.1	Si las credenciales ingresadas no corresponden a credenciales genéricas se redirecciona al dashboard principal	6.1	En caso que algún parámetro del cambio de credenciales sea incorrecto se resaltará en color rojo el campo faltan o erróneo y no permitirá avanzar al paso 7								
Paso	Acción																
2.1	Si el correo o la contraseña son incorrectos muestra un mensaje genérico de credenciales erróneas y retorna al formulario de autenticación paso 1																
3.1	Si las credenciales ingresadas no corresponden a credenciales genéricas se redirecciona al dashboard principal																
6.1	En caso que algún parámetro del cambio de credenciales sea incorrecto se resaltará en color rojo el campo faltan o erróneo y no permitirá avanzar al paso 7																
Postcondición	Cambio exitoso de credenciales genéricas																

Fuente: Elaboración Propia

En la siguiente Ilustración 27 se evidencia el caso de uso de cambio de credenciales por defecto.

Ilustración 27. Caso de Uso Cambio de Credenciales por Defecto



Fuente: Elaboración Propia

3.2.4.4 Formato Caso de Uso Estado Dispositivo Raspberry Pi

Tabla 28. Caso de Uso Estado Dispositivo Raspberry Pi

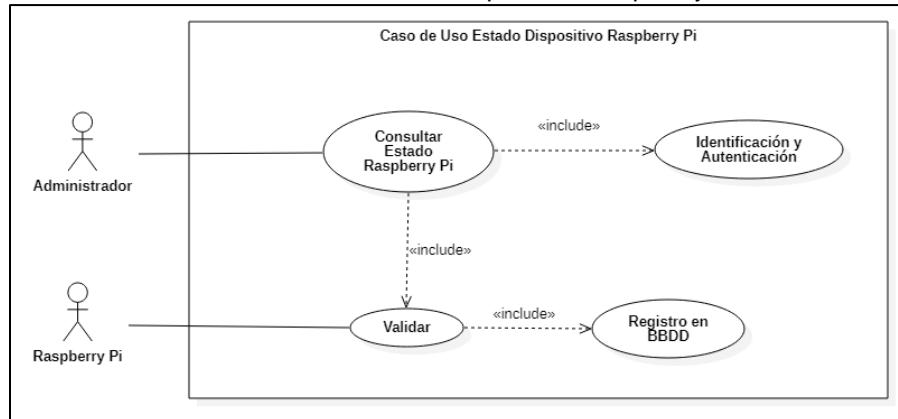
Caso de Uso	Estado Dispositivo Raspberry Pi						
Actor	Administrador						
Tipo	Secundario						
Descripción	Permite realizar un monitoreo en tiempo real a nivel de red sobre la Raspberry Pi, garantizando que procesos posteriores se lleven de forma adecuada.						
Precondición	El usuario se encuentra autenticado y autorizado para ingresar a este apartado						
Flujo Básico	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El administrador ingresa al módulo estado de dispositivo</td> </tr> <tr> <td>2</td> <td>El sistema trae la información del estado del dispositivo Raspberry Pi</td> </tr> </tbody> </table>	Paso	Acción	1	El administrador ingresa al módulo estado de dispositivo	2	El sistema trae la información del estado del dispositivo Raspberry Pi
Paso	Acción						
1	El administrador ingresa al módulo estado de dispositivo						
2	El sistema trae la información del estado del dispositivo Raspberry Pi						

	3	El sistema guarda a nivel de base de datos el estado actual de la Raspberry Pi por si en algún momento el administrador desea consultarla
	3	Finaliza el caso de uso
Flujo Alternativo	Paso	Acción
	2.1	Sí el sistema detecta algún error al momento de traer la información no se muestra al usuario resultados
Postcondición		Guarda a nivel de base de datos el estado actual para posteriores consultas

Fuente: Elaboración Propia

En la siguiente Ilustración 28 se evidencia el caso de uso de estado dispositivo Raspberry Pi.

Ilustración 28. Caso de Uso Estado Dispositivo Raspberry Pi



Fuente: Elaboración Propia

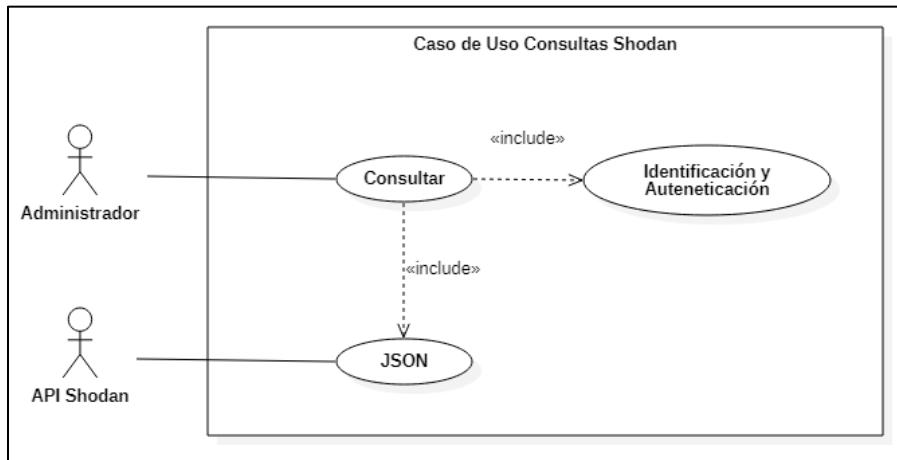
3.2.4.5 Formato Caso de Uso Consultas Shodan

Tabla 29. Caso de Uso Consultas Shodan

Caso de Uso	Consultas Shodan										
Actor	Administrador										
Tipo	Primario										
Descripción	Permite realizar por medio de la API de Shodan consultas a dispositivos expuestos en internet										
Precondición	API Key implementada en software										
Flujo Básico	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El administrador ingresa al módulo de Shodan</td> </tr> <tr> <td>2</td> <td>El administrador ingresa un parámetro de búsqueda como puertos, nombre de sistema, servicio, ubicación, etc.</td> </tr> <tr> <td>3</td> <td>El sistema por medio de la API de Shodan realiza una búsqueda y trae resultados asociados a los parámetros solicitados.</td> </tr> <tr> <td>4</td> <td>Finaliza el caso de uso</td> </tr> </tbody> </table>	Paso	Acción	1	El administrador ingresa al módulo de Shodan	2	El administrador ingresa un parámetro de búsqueda como puertos, nombre de sistema, servicio, ubicación, etc.	3	El sistema por medio de la API de Shodan realiza una búsqueda y trae resultados asociados a los parámetros solicitados.	4	Finaliza el caso de uso
Paso	Acción										
1	El administrador ingresa al módulo de Shodan										
2	El administrador ingresa un parámetro de búsqueda como puertos, nombre de sistema, servicio, ubicación, etc.										
3	El sistema por medio de la API de Shodan realiza una búsqueda y trae resultados asociados a los parámetros solicitados.										
4	Finaliza el caso de uso										
Flujo Alternativo	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>3.1</td> <td>Sí se detecta algún error o problema al momento de consultar la API de Shodan se muestra un error de configuración</td> </tr> </tbody> </table>	Paso	Acción	3.1	Sí se detecta algún error o problema al momento de consultar la API de Shodan se muestra un error de configuración						
Paso	Acción										
3.1	Sí se detecta algún error o problema al momento de consultar la API de Shodan se muestra un error de configuración										
Postcondición	Muestra una lista de resultados										

Fuente: Elaboración Propia

En la siguiente Ilustración 29 se evidencia el caso de uso de consultas Shodan.

Ilustración 29. Caso de Uso Consultas Shodan

Fuente: Elaboración Propia

3.2.4.6 Formato Caso de Uso para Crear Escaneo de Descubrimiento de Objetos IoT

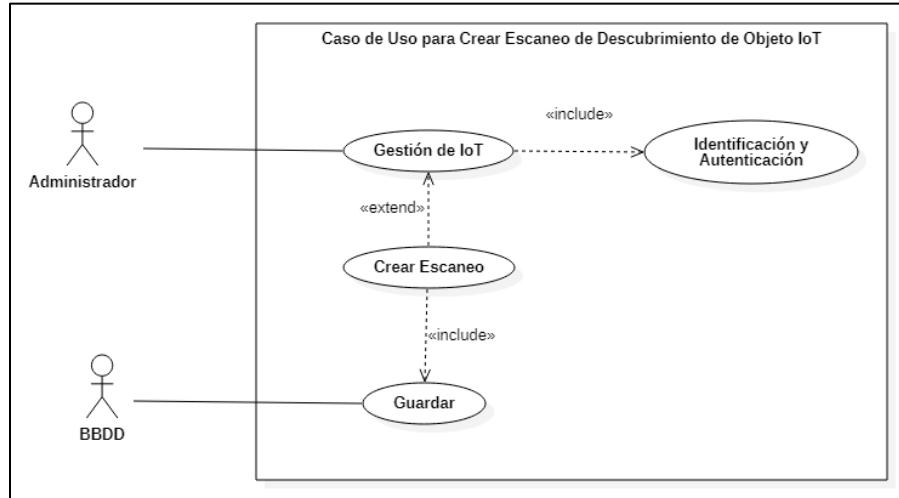
Tabla 30. Caso de Uso para Crear Escaneo de Descubrimiento de Objetos IoT

Caso de Uso	Crear Escaneo de Descubrimiento de Objetos IoT														
Actor	Administrador														
Tipo	Primario														
Descripción	Permite hacer la creación de un escaneo para el descubrimiento de puertos, servicios y estados sobre uno o varios objetos IoT														
Precondición	El usuario debe estar autenticado														
Flujo Básico	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El administrador ingresa al módulo de gestión de IoT</td></tr> <tr> <td>2</td><td>El administrador hace clic en la opción "Nuevo Escaneo"</td></tr> <tr> <td>3</td><td>El administrador ingresa el nombre, descripción y dirección(es) IP de los objeto(s) IoT</td></tr> <tr> <td>4</td><td>El administrador hace clic en la opción "Crear" para confirmar creación y guardar la información a nivel de base de datos</td></tr> <tr> <td>5</td><td>El sistema envía una notificación satisfactoria y agrega a la lista de escaneos el creado</td></tr> <tr> <td>6</td><td>Finaliza el caso de uso</td></tr> </tbody> </table>	Paso	Acción	1	El administrador ingresa al módulo de gestión de IoT	2	El administrador hace clic en la opción "Nuevo Escaneo"	3	El administrador ingresa el nombre, descripción y dirección(es) IP de los objeto(s) IoT	4	El administrador hace clic en la opción "Crear" para confirmar creación y guardar la información a nivel de base de datos	5	El sistema envía una notificación satisfactoria y agrega a la lista de escaneos el creado	6	Finaliza el caso de uso
Paso	Acción														
1	El administrador ingresa al módulo de gestión de IoT														
2	El administrador hace clic en la opción "Nuevo Escaneo"														
3	El administrador ingresa el nombre, descripción y dirección(es) IP de los objeto(s) IoT														
4	El administrador hace clic en la opción "Crear" para confirmar creación y guardar la información a nivel de base de datos														
5	El sistema envía una notificación satisfactoria y agrega a la lista de escaneos el creado														
6	Finaliza el caso de uso														
Flujo Alternativo	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>4.1</td><td>En caso que el usuario haga clic en "Cancelar" no se crea ni se guarda a nivel de base de datos el escaneo</td></tr> <tr> <td>5.1</td><td>Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no se crea el escaneo</td></tr> </tbody> </table>	Paso	Acción	4.1	En caso que el usuario haga clic en "Cancelar" no se crea ni se guarda a nivel de base de datos el escaneo	5.1	Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no se crea el escaneo								
Paso	Acción														
4.1	En caso que el usuario haga clic en "Cancelar" no se crea ni se guarda a nivel de base de datos el escaneo														
5.1	Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no se crea el escaneo														
Postcondición	Guarda escaneo creado														

Fuente: Elaboración Propia

En la siguiente Ilustración 30 se evidencia el caso de uso para crear escaneo de descubrimiento de objetos IoT.

Ilustración 30. Caso de Uso para Crear Escaneo de Descubrimiento de Objetos IoT



Fuente: Elaboración Propia

3.2.4.7 Formato Caso de Uso para Modificar Escaneo de Descubrimiento de Objetos IoT

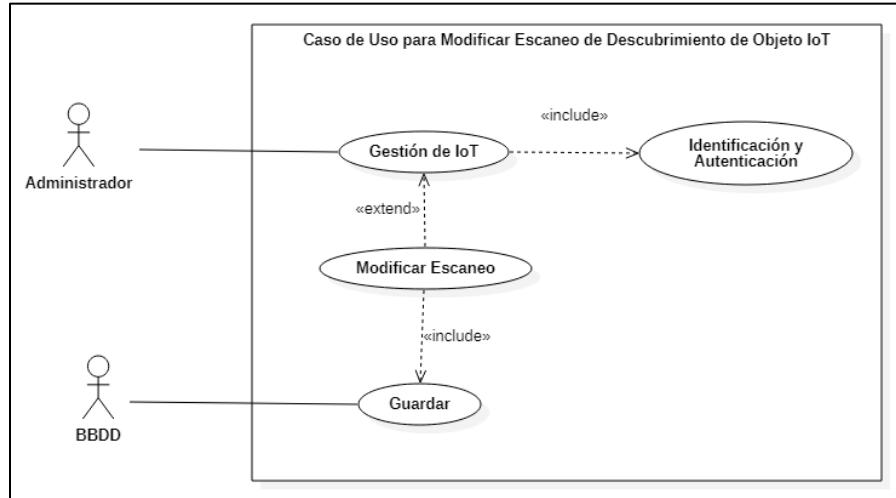
Tabla 31. Caso de Uso para Modificar Escaneo de Descubrimiento de Objetos IoT

Caso de Uso	Modificar Escaneo de Descubrimiento de Objetos IoT														
Actor	Administrador														
Tipo	Primario														
Descripción	Permite hacer la modificación de un escaneo para el descubrimiento de puertos, servicios y estados sobre uno o varios objetos IoT														
Precondición	El usuario debe estar autenticado y debe existir un escaneo previamente creado														
Flujo Básico	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El administrador ingresa al módulo de gestión de IoT</td></tr> <tr> <td>2</td><td>El administrador hace clic en la opción "Modificar" del escaneo sobre el cual desea tomar acción</td></tr> <tr> <td>3</td><td>El administrador puede modificar el nombre, descripción y dirección(es) IP de los objeto(s) IoT, dependiendo de la necesidad</td></tr> <tr> <td>4</td><td>El administrador hace clic en la opción "Modificar" para confirmar modificación y guardar la información a nivel de base de datos</td></tr> <tr> <td>5</td><td>El sistema envía una notificación satisfactoria y actualiza en la lista de escaneos sobre el cual se tomó la acción</td></tr> <tr> <td>6</td><td>Finaliza el caso de uso</td></tr> </tbody> </table>	Paso	Acción	1	El administrador ingresa al módulo de gestión de IoT	2	El administrador hace clic en la opción "Modificar" del escaneo sobre el cual desea tomar acción	3	El administrador puede modificar el nombre, descripción y dirección(es) IP de los objeto(s) IoT, dependiendo de la necesidad	4	El administrador hace clic en la opción "Modificar" para confirmar modificación y guardar la información a nivel de base de datos	5	El sistema envía una notificación satisfactoria y actualiza en la lista de escaneos sobre el cual se tomó la acción	6	Finaliza el caso de uso
Paso	Acción														
1	El administrador ingresa al módulo de gestión de IoT														
2	El administrador hace clic en la opción "Modificar" del escaneo sobre el cual desea tomar acción														
3	El administrador puede modificar el nombre, descripción y dirección(es) IP de los objeto(s) IoT, dependiendo de la necesidad														
4	El administrador hace clic en la opción "Modificar" para confirmar modificación y guardar la información a nivel de base de datos														
5	El sistema envía una notificación satisfactoria y actualiza en la lista de escaneos sobre el cual se tomó la acción														
6	Finaliza el caso de uso														
Flujo Alternativo	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>4.1</td><td>En caso que el usuario haga clic en "Cancelar" no se actualiza ni se guarda a nivel de base de datos las modificaciones del escaneo</td></tr> <tr> <td>5.1</td><td>Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no se modifica el escaneo</td></tr> </tbody> </table>	Paso	Acción	4.1	En caso que el usuario haga clic en "Cancelar" no se actualiza ni se guarda a nivel de base de datos las modificaciones del escaneo	5.1	Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no se modifica el escaneo								
Paso	Acción														
4.1	En caso que el usuario haga clic en "Cancelar" no se actualiza ni se guarda a nivel de base de datos las modificaciones del escaneo														
5.1	Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no se modifica el escaneo														
Postcondición	Guarda escaneo modificado														

Fuente: Elaboración Propia

En la siguiente Ilustración 31 se evidencia el caso de uso para modificar escaneo de descubrimiento de objetos IoT.

Ilustración 31. Caso de Uso para Modificar Escaneo de Descubrimiento de Objetos IoT



Fuente: Elaboración Propia

3.2.4.8 Formato Caso de Uso para Eliminar Escaneo de Descubrimiento de Objetos IoT

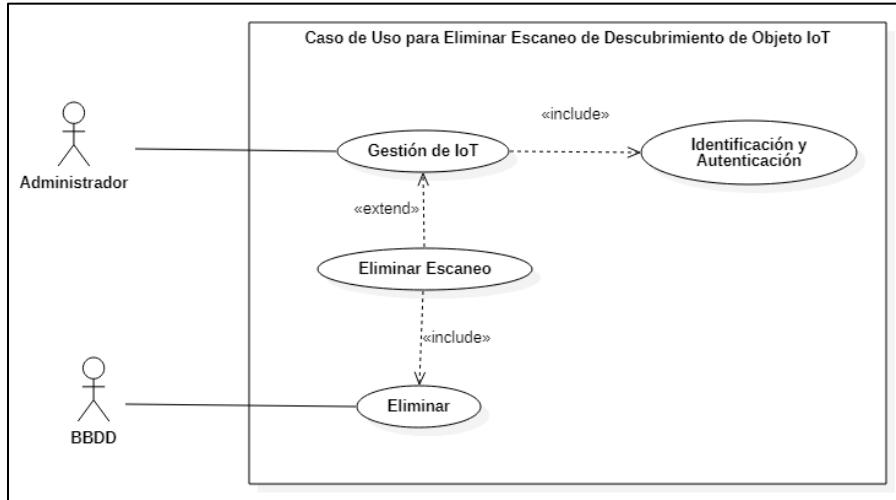
Tabla 32. Caso de Uso para Eliminar Escaneo de Descubrimiento de Objetos IoT

Caso de Uso	Eliminar Escaneo de Descubrimiento de Objetos IoT												
Actor	Administrador												
Tipo	Primario												
Descripción	Permite hacer la eliminación de un escaneo para el descubrimiento de puertos, servicios y estados sobre uno o varios objetos IoT												
Precondición	El usuario debe estar autenticado y debe existir un escaneo previamente creado												
Flujo Básico	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El administrador ingresa al módulo de gestión de IoT</td></tr> <tr> <td>2</td><td>El administrador hace clic en la opción “Eliminar” del escaneo sobre el cual desea tomar acción</td></tr> <tr> <td>3</td><td>En el dialogo de confirmación de eliminación de escaneo, el administrador hace clic en “Si, eliminar”</td></tr> <tr> <td>4</td><td>El sistema envía una notificación satisfactoria y elimina de la lista de escaneos el cual se tomó la acción</td></tr> <tr> <td>5</td><td>Finaliza el caso de uso</td></tr> </tbody> </table>	Paso	Acción	1	El administrador ingresa al módulo de gestión de IoT	2	El administrador hace clic en la opción “Eliminar” del escaneo sobre el cual desea tomar acción	3	En el dialogo de confirmación de eliminación de escaneo, el administrador hace clic en “Si, eliminar”	4	El sistema envía una notificación satisfactoria y elimina de la lista de escaneos el cual se tomó la acción	5	Finaliza el caso de uso
Paso	Acción												
1	El administrador ingresa al módulo de gestión de IoT												
2	El administrador hace clic en la opción “Eliminar” del escaneo sobre el cual desea tomar acción												
3	En el dialogo de confirmación de eliminación de escaneo, el administrador hace clic en “Si, eliminar”												
4	El sistema envía una notificación satisfactoria y elimina de la lista de escaneos el cual se tomó la acción												
5	Finaliza el caso de uso												
Flujo Alternativo	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>3.1</td><td>En caso que el usuario haga clic en “Cancelar” no se elimina y aún se mantiene guardado a nivel de base de datos el escaneo</td></tr> <tr> <td>4.1</td><td>Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no se elimina el escaneo</td></tr> </tbody> </table>	Paso	Acción	3.1	En caso que el usuario haga clic en “Cancelar” no se elimina y aún se mantiene guardado a nivel de base de datos el escaneo	4.1	Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no se elimina el escaneo						
Paso	Acción												
3.1	En caso que el usuario haga clic en “Cancelar” no se elimina y aún se mantiene guardado a nivel de base de datos el escaneo												
4.1	Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no se elimina el escaneo												
Postcondición	Elimina escaneo												

Fuente: Elaboración Propia

En la siguiente Ilustración 32 se evidencia el caso de uso para eliminar escaneo de descubrimiento de objetos IoT.

Ilustración 32. Caso de Uso para Eliminar Escaneo de Descubrimiento de Objetos IoT



Fuente: Elaboración Propia

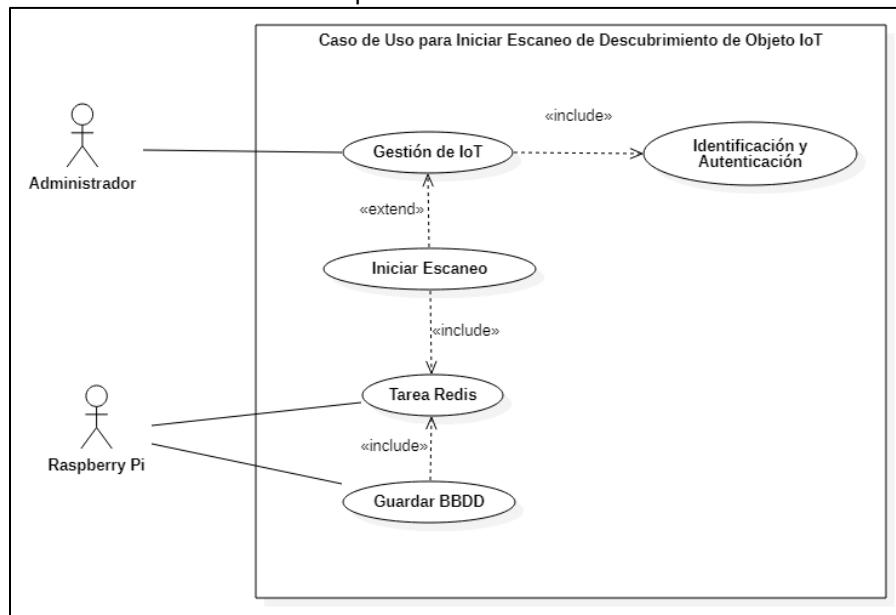
3.2.4.9 Formato Caso de Uso para Iniciar Escaneo de Descubrimiento de Objetos IoT

Tabla 33. Caso de Uso para Iniciar Escaneo de Descubrimiento de Objetos IoT

Caso de Uso	Iniciar Escaneo de Descubrimiento de Objetos IoT																
Actor	Administrador																
Tipo	Primario																
Descripción	Permite iniciar un escaneo para el descubrimiento de puertos, servicios y estados sobre uno o varios objetos IoT																
Precondición	El usuario debe estar autenticado, debe existir un escaneo previamente creado y debe existir una tarea Redis a la escucha																
Flujo Básico	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El administrador ingresa al módulo de gestión de IoT</td></tr> <tr> <td>2</td><td>El administrador hace clic en la opción “Iniciar Escaneo” del escaneo sobre el cual desea tomar acción</td></tr> <tr> <td>3</td><td>La tarea Redis en escucha es usada para iniciar el escaneo</td></tr> <tr> <td>4</td><td>El sistema envía una notificación satisfactoria de inicio</td></tr> <tr> <td>5</td><td>El sistema establece comunicación con el objeto IoT y por medio de protocolos de red identifica puertos y servicios activos</td></tr> <tr> <td>6</td><td>El sistema clasifica y guarda la información relevante a nivel de base de datos</td></tr> <tr> <td>7</td><td>Finaliza el caso de uso</td></tr> </tbody> </table>	Paso	Acción	1	El administrador ingresa al módulo de gestión de IoT	2	El administrador hace clic en la opción “Iniciar Escaneo” del escaneo sobre el cual desea tomar acción	3	La tarea Redis en escucha es usada para iniciar el escaneo	4	El sistema envía una notificación satisfactoria de inicio	5	El sistema establece comunicación con el objeto IoT y por medio de protocolos de red identifica puertos y servicios activos	6	El sistema clasifica y guarda la información relevante a nivel de base de datos	7	Finaliza el caso de uso
Paso	Acción																
1	El administrador ingresa al módulo de gestión de IoT																
2	El administrador hace clic en la opción “Iniciar Escaneo” del escaneo sobre el cual desea tomar acción																
3	La tarea Redis en escucha es usada para iniciar el escaneo																
4	El sistema envía una notificación satisfactoria de inicio																
5	El sistema establece comunicación con el objeto IoT y por medio de protocolos de red identifica puertos y servicios activos																
6	El sistema clasifica y guarda la información relevante a nivel de base de datos																
7	Finaliza el caso de uso																
Flujo Alternativo	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>3.1</td><td>No hay tarea de Redis ejecutándose a la escucha, generando error</td></tr> <tr> <td>4.1</td><td>Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error.</td></tr> <tr> <td>5.1</td><td>El sistema no logra establecer comunicación con el objeto IoT debido a que no se encuentra activo o problemas de red</td></tr> <tr> <td>6.1</td><td>El sistema no logra guardar o identificar información relevante a nivel de base de datos</td></tr> </tbody> </table>	Paso	Acción	3.1	No hay tarea de Redis ejecutándose a la escucha, generando error	4.1	Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error.	5.1	El sistema no logra establecer comunicación con el objeto IoT debido a que no se encuentra activo o problemas de red	6.1	El sistema no logra guardar o identificar información relevante a nivel de base de datos						
Paso	Acción																
3.1	No hay tarea de Redis ejecutándose a la escucha, generando error																
4.1	Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error.																
5.1	El sistema no logra establecer comunicación con el objeto IoT debido a que no se encuentra activo o problemas de red																
6.1	El sistema no logra guardar o identificar información relevante a nivel de base de datos																
Postcondición	Inicia escaneo																

Fuente: Elaboración Propia

En la siguiente Ilustración 33 se evidencia el caso de uso para iniciar escaneo de descubrimiento de objetos IoT.

Ilustración 33. Caso de Uso para Iniciar Escaneo de Descubrimiento de Objetos IoT

Fuente: Elaboración Propia

3.2.4.10 Formato Caso de Uso para Mostrar el Resultado del Escaneo de Descubrimiento de Objetos IoT

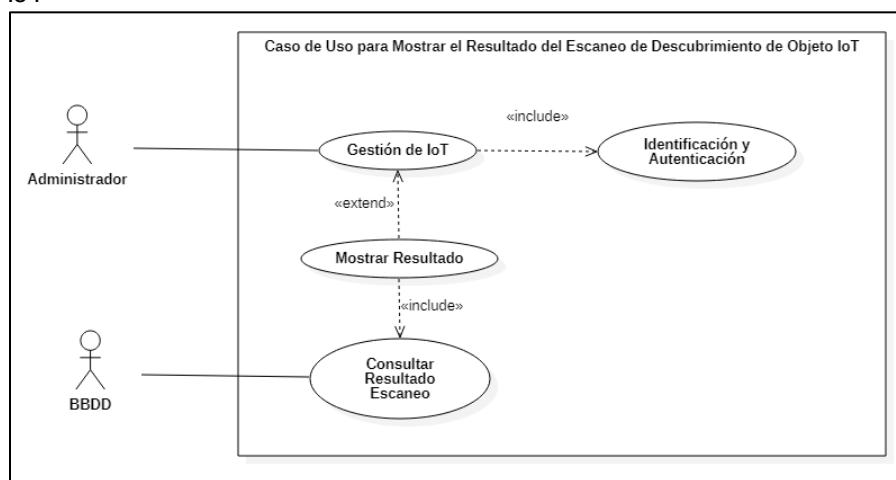
Tabla 34. Caso de Uso para Mostrar el Resultado del Escaneo de Descubrimiento de Objetos IoT

Caso de Uso	Mostrar el Resultado del Escaneo de Descubrimiento de Objetos IoT	
Actor	Administrador	
Tipo	Primario	
Descripción	Permite visualizar el resultado de un escaneo para el descubrimiento de puertos, servicios y estados sobre uno o varios objetos IoT	
Precondición	El usuario debe estar autenticado y debe haber terminado con éxito un escaneo	
Flujo Básico	Paso	Acción
	1	El administrador ingresa al módulo de gestión de IoT
	2	El administrador hace clic en la opción “Mostrar Resultado” del escaneo sobre el cual desea tomar acción
	3	El sistema recupera la información recolectada a nivel de base de datos
	4	El sistema muestra de forma organizada información referente al estado del objeto IoT, OS, dirección IPv4, puerto, servicios, etc.
Flujo Alternativo	Paso	Acción
	3.1	Si en el escaneo no se guardó información a nivel de base datos no se recolecta información
	4.1	Si en el escaneo no se guardó información a nivel de base datos no se muestra información al usuario
Postcondición	Mostrar resultado de escaneo	

Fuente: Elaboración Propia

En la siguiente Ilustración 34 se evidencia el caso de uso para mostrar el resultado del escaneo de descubrimiento de objetos IoT.

Ilustración 34. Caso de Uso para Mostrar el Resultado del Escaneo de Descubrimiento de Objetos IoT



Fuente: Elaboración Propia

3.2.4.11 Formato Caso de Uso Seleccionar Objetivos para Escaneo de Vulnerabilidades

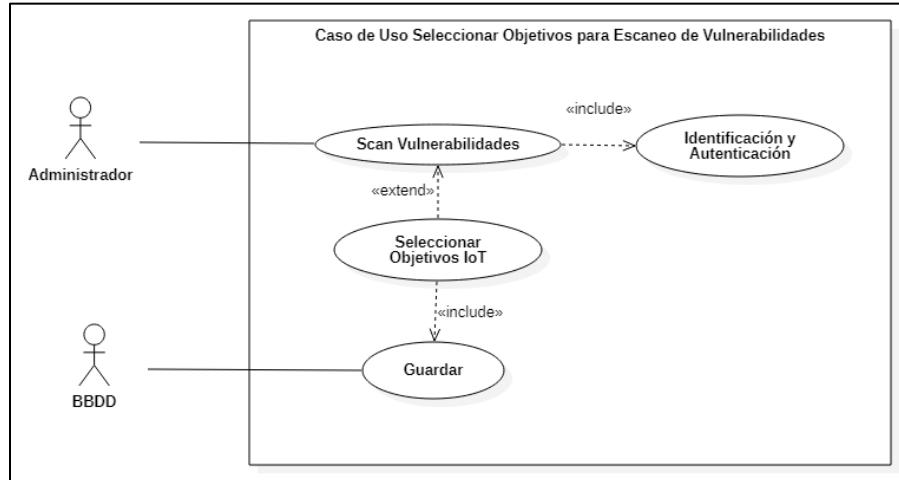
Tabla 35. Caso de Uso Seleccionar Objetivos para Escaneo de Vulnerabilidades

Caso de Uso	Seleccionar Objetivos para Escaneo de Vulnerabilidades																
Actor	Administrador																
Tipo	Primario																
Descripción	Permite seleccionar de la lista de objetos IoT encontrados activos los que se deseé para hacer el análisis de vulnerabilidades.																
Precondición	El usuario debe estar autenticado y debe haber objetos IoT activos identificados en el escaneo de descubrimiento.																
Flujo Básico	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El administrador ingresa al módulo scan vulnerabilidades</td></tr> <tr> <td>2</td><td>El administrador hace clic en la opción “Seleccionar Objetivos” del escaneo sobre el cual desea tomar acción</td></tr> <tr> <td>3</td><td>El sistema recupera la lista de los objetos IoT activos existentes detectados</td></tr> <tr> <td>4</td><td>El administrador selecciona el objeto IoT u objetos IoT a los cuales desea hacer el escaneo de vulnerabilidades</td></tr> <tr> <td>5</td><td>El administrador guarda la lista</td></tr> <tr> <td>6</td><td>El sistema lo guarda a nivel de base de datos y envía una notificación satisfactoria</td></tr> <tr> <td>7</td><td>Finaliza el caso de uso</td></tr> </tbody> </table>	Paso	Acción	1	El administrador ingresa al módulo scan vulnerabilidades	2	El administrador hace clic en la opción “Seleccionar Objetivos” del escaneo sobre el cual desea tomar acción	3	El sistema recupera la lista de los objetos IoT activos existentes detectados	4	El administrador selecciona el objeto IoT u objetos IoT a los cuales desea hacer el escaneo de vulnerabilidades	5	El administrador guarda la lista	6	El sistema lo guarda a nivel de base de datos y envía una notificación satisfactoria	7	Finaliza el caso de uso
Paso	Acción																
1	El administrador ingresa al módulo scan vulnerabilidades																
2	El administrador hace clic en la opción “Seleccionar Objetivos” del escaneo sobre el cual desea tomar acción																
3	El sistema recupera la lista de los objetos IoT activos existentes detectados																
4	El administrador selecciona el objeto IoT u objetos IoT a los cuales desea hacer el escaneo de vulnerabilidades																
5	El administrador guarda la lista																
6	El sistema lo guarda a nivel de base de datos y envía una notificación satisfactoria																
7	Finaliza el caso de uso																
Flujo Alternativo	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>3.1</td><td>Si no se ha ejecutado el escaneo para descubrir objetos IoT no se va a mostrar ningún valor</td></tr> <tr> <td>3.2</td><td>Si no se ha encontrado ningún objeto IoT activo no se va a mostrar ningún valor</td></tr> <tr> <td>5.1</td><td>El administrador cierra el dialogo de selección y no se guarda la información</td></tr> <tr> <td>6.1</td><td>Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no guarda la información</td></tr> </tbody> </table>	Paso	Acción	3.1	Si no se ha ejecutado el escaneo para descubrir objetos IoT no se va a mostrar ningún valor	3.2	Si no se ha encontrado ningún objeto IoT activo no se va a mostrar ningún valor	5.1	El administrador cierra el dialogo de selección y no se guarda la información	6.1	Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no guarda la información						
Paso	Acción																
3.1	Si no se ha ejecutado el escaneo para descubrir objetos IoT no se va a mostrar ningún valor																
3.2	Si no se ha encontrado ningún objeto IoT activo no se va a mostrar ningún valor																
5.1	El administrador cierra el dialogo de selección y no se guarda la información																
6.1	Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no guarda la información																
Postcondición	Seleccionar objetivos IoT para escaneo de vulnerabilidades																

Fuente: Elaboración Propia

En la siguiente Ilustración 35 se evidencia el caso de uso seleccionar objetivos para escaneo de vulnerabilidades.

Ilustración 35. Caso de Uso Seleccionar Objetivos para Escaneo de Vulnerabilidades



Fuente: Elaboración Propia

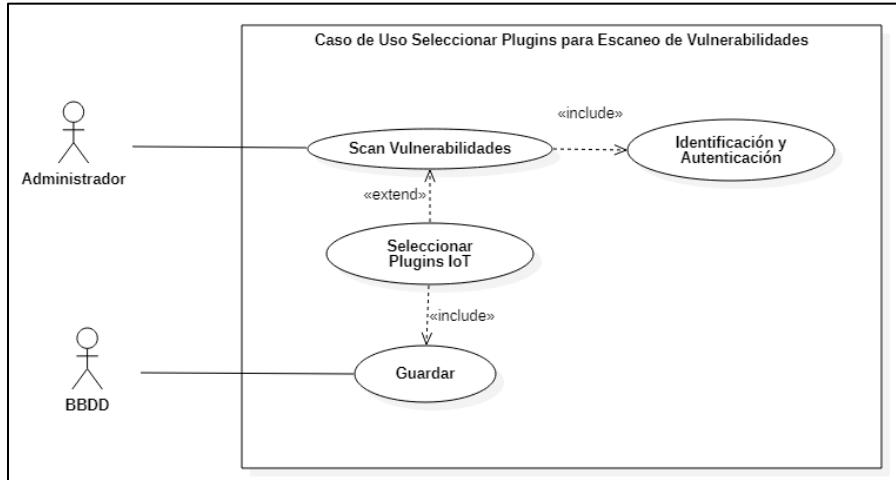
3.2.4.12 Formato Caso de Uso Seleccionar Plugins para Escaneo de Vulnerabilidades

Tabla 36. Caso de Uso Seleccionar Plugins para Escaneo de Vulnerabilidades

Caso de Uso	Seleccionar Plugins para Escaneo de Vulnerabilidades																
Actor	Administrador																
Tipo	Primario																
Descripción	Permite seleccionar de la lista de plugins los que se desee para hacer el análisis de vulnerabilidades.																
Precondición	El usuario debe estar autenticado y debe haber objetos IoT previamente seleccionados para análisis de vulnerabilidades																
Flujo Básico	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El administrador ingresa al módulo scan vulnerabilidades</td></tr> <tr> <td>2</td><td>El administrador hace clic en la opción "Seleccionar Plugins" del escaneo sobre el cual desea tomar acción</td></tr> <tr> <td>3</td><td>De acuerdo a los puertos y servicios descubiertos sobre los objetos IoT previamente seleccionados, el sistema despliega la lista de plugins para el análisis de vulnerabilidades</td></tr> <tr> <td>4</td><td>El administrador selecciona el plugin o lista de plugins los cuales se harán uso en el escaneo de vulnerabilidades</td></tr> <tr> <td>5</td><td>El administrador guarda la lista</td></tr> <tr> <td>6</td><td>El sistema lo guarda a nivel de base de datos y envía una notificación satisfactoria</td></tr> <tr> <td>7</td><td>Finaliza el caso de uso</td></tr> </tbody> </table>	Paso	Acción	1	El administrador ingresa al módulo scan vulnerabilidades	2	El administrador hace clic en la opción "Seleccionar Plugins" del escaneo sobre el cual desea tomar acción	3	De acuerdo a los puertos y servicios descubiertos sobre los objetos IoT previamente seleccionados, el sistema despliega la lista de plugins para el análisis de vulnerabilidades	4	El administrador selecciona el plugin o lista de plugins los cuales se harán uso en el escaneo de vulnerabilidades	5	El administrador guarda la lista	6	El sistema lo guarda a nivel de base de datos y envía una notificación satisfactoria	7	Finaliza el caso de uso
Paso	Acción																
1	El administrador ingresa al módulo scan vulnerabilidades																
2	El administrador hace clic en la opción "Seleccionar Plugins" del escaneo sobre el cual desea tomar acción																
3	De acuerdo a los puertos y servicios descubiertos sobre los objetos IoT previamente seleccionados, el sistema despliega la lista de plugins para el análisis de vulnerabilidades																
4	El administrador selecciona el plugin o lista de plugins los cuales se harán uso en el escaneo de vulnerabilidades																
5	El administrador guarda la lista																
6	El sistema lo guarda a nivel de base de datos y envía una notificación satisfactoria																
7	Finaliza el caso de uso																
Flujo Alternativo	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>3.1</td><td>Si no se ha seleccionado el objeto o lista de objetos IoT el sistema no se va a mostrar ningún valor</td></tr> <tr> <td>5.1</td><td>El administrador cierra el dialogo de selección y no se guarda la información</td></tr> <tr> <td>6.1</td><td>Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no guarda la información</td></tr> </tbody> </table>	Paso	Acción	3.1	Si no se ha seleccionado el objeto o lista de objetos IoT el sistema no se va a mostrar ningún valor	5.1	El administrador cierra el dialogo de selección y no se guarda la información	6.1	Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no guarda la información								
Paso	Acción																
3.1	Si no se ha seleccionado el objeto o lista de objetos IoT el sistema no se va a mostrar ningún valor																
5.1	El administrador cierra el dialogo de selección y no se guarda la información																
6.1	Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no guarda la información																
Postcondición	Seleccionar plugins para escaneo de vulnerabilidades																

Fuente: Elaboración Propia

En la siguiente Ilustración 36 se evidencia el caso de uso seleccionar plugins para escaneo de vulnerabilidades.

Ilustración 36. Caso de Uso Seleccionar Plugins para Escaneo de Vulnerabilidades

Fuente: Elaboración Propia

3.2.4.13 Formato Caso de Uso Iniciar Escaneo de Vulnerabilidades

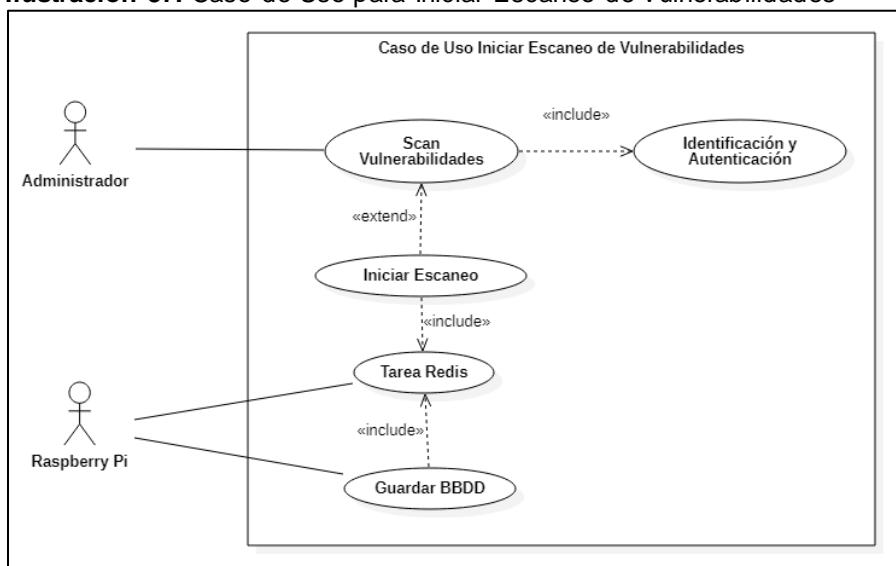
Tabla 37. Caso de Uso Iniciar Escaneo de Vulnerabilidades

Caso de Uso	Iniciar Escaneo de Vulnerabilidades																		
Actor	Administrador																		
Tipo	Primario																		
Descripción	Permite iniciar el escaneo de vulnerabilidades sobre objetos IoT																		
Precondición	El usuario debe estar autenticado, debe haber una tarea Redis a la escucha y deben estar seleccionados objetos IoT activos y plugins.																		
Flujo Básico	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El administrador ingresa al módulo de scan vulnerabilidades</td></tr> <tr> <td>2</td><td>El administrador hace clic en la opción “Iniciar Escaneo” del escaneo sobre el cual desea tomar acción</td></tr> <tr> <td>3</td><td>El sistema pregunta si se han seleccionado los objetos IoT y los plugins, el administrado confirma “Si, Iniciar”</td></tr> <tr> <td>4</td><td>La tarea Redis en escucha es usada para iniciar el escaneo de vulnerabilidades</td></tr> <tr> <td>5</td><td>El sistema envía una notificación satisfactoria de inicio</td></tr> <tr> <td>6</td><td>El sistema establece comunicación con el objeto IoT y por medio de diversos scripts – exploits – verifica si es vulnerable a alguna vulnerabilidad en algún servicio o puerto</td></tr> <tr> <td>7</td><td>El sistema clasifica y guarda la información de vulnerabilidades confirmadas a nivel de base de datos</td></tr> <tr> <td>8</td><td>Finaliza el caso de uso</td></tr> </tbody> </table>	Paso	Acción	1	El administrador ingresa al módulo de scan vulnerabilidades	2	El administrador hace clic en la opción “Iniciar Escaneo” del escaneo sobre el cual desea tomar acción	3	El sistema pregunta si se han seleccionado los objetos IoT y los plugins, el administrado confirma “Si, Iniciar”	4	La tarea Redis en escucha es usada para iniciar el escaneo de vulnerabilidades	5	El sistema envía una notificación satisfactoria de inicio	6	El sistema establece comunicación con el objeto IoT y por medio de diversos scripts – exploits – verifica si es vulnerable a alguna vulnerabilidad en algún servicio o puerto	7	El sistema clasifica y guarda la información de vulnerabilidades confirmadas a nivel de base de datos	8	Finaliza el caso de uso
Paso	Acción																		
1	El administrador ingresa al módulo de scan vulnerabilidades																		
2	El administrador hace clic en la opción “Iniciar Escaneo” del escaneo sobre el cual desea tomar acción																		
3	El sistema pregunta si se han seleccionado los objetos IoT y los plugins, el administrado confirma “Si, Iniciar”																		
4	La tarea Redis en escucha es usada para iniciar el escaneo de vulnerabilidades																		
5	El sistema envía una notificación satisfactoria de inicio																		
6	El sistema establece comunicación con el objeto IoT y por medio de diversos scripts – exploits – verifica si es vulnerable a alguna vulnerabilidad en algún servicio o puerto																		
7	El sistema clasifica y guarda la información de vulnerabilidades confirmadas a nivel de base de datos																		
8	Finaliza el caso de uso																		
Flujo Alternativo	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>3.1</td><td>En caso que el usuario haga clic en “Cancelar” no se inicia el escaneo de vulnerabilidades</td></tr> <tr> <td>4.1</td><td>No hay tarea de Redis ejecutándose a la escucha, generando error</td></tr> <tr> <td>5.1</td><td>Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error.</td></tr> <tr> <td>6.1</td><td>El sistema no logra establecer comunicación con el objeto IoT debido a que no se encuentra activo o problemas de red</td></tr> <tr> <td>7.1</td><td>El sistema no logra guardar o identificar información relevante a nivel de base de datos</td></tr> </tbody> </table>	Paso	Acción	3.1	En caso que el usuario haga clic en “Cancelar” no se inicia el escaneo de vulnerabilidades	4.1	No hay tarea de Redis ejecutándose a la escucha, generando error	5.1	Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error.	6.1	El sistema no logra establecer comunicación con el objeto IoT debido a que no se encuentra activo o problemas de red	7.1	El sistema no logra guardar o identificar información relevante a nivel de base de datos						
Paso	Acción																		
3.1	En caso que el usuario haga clic en “Cancelar” no se inicia el escaneo de vulnerabilidades																		
4.1	No hay tarea de Redis ejecutándose a la escucha, generando error																		
5.1	Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error.																		
6.1	El sistema no logra establecer comunicación con el objeto IoT debido a que no se encuentra activo o problemas de red																		
7.1	El sistema no logra guardar o identificar información relevante a nivel de base de datos																		
Postcondición	Inicia escaneo																		

Fuente: Elaboración Propia

En la siguiente Ilustración 37 se evidencia el caso de uso para iniciar escaneo de vulnerabilidades.

Ilustración 37. Caso de Uso para Iniciar Escaneo de Vulnerabilidades



Fuente: Elaboración Propia

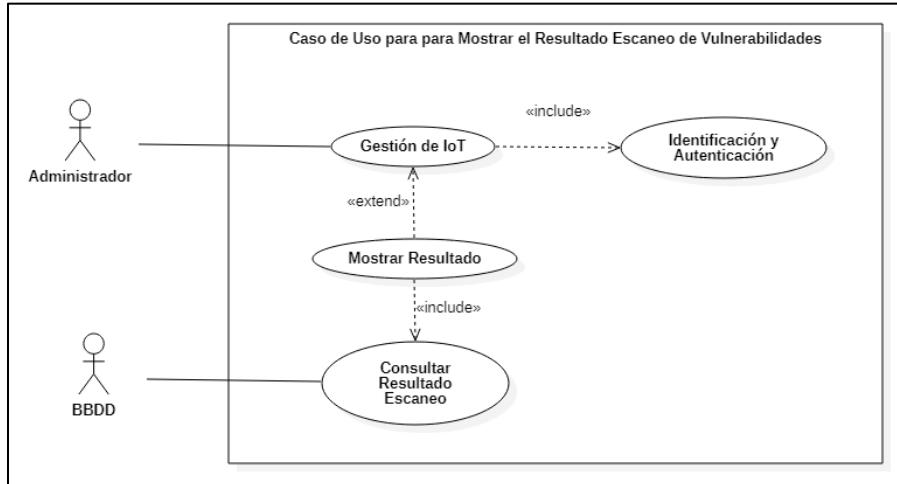
3.2.4.14 Formato Caso de Uso para Mostrar el Resultado Escaneo de Vulnerabilidades

Tabla 38. Caso de Uso para Mostrar el Resultado Escaneo de Vulnerabilidades

Caso de Uso	Mostrar el Resultado Escaneo de Vulnerabilidades	
Actor	Administrador	
Tipo	Primario	
Descripción	Permite visualizar de forma organizada en tablas y gráficas el resultado de un escaneo de vulnerabilidades sobre un objeto IoT	
Precondición	El usuario debe estar autenticado y debe haber terminado con éxito un escaneo de vulnerabilidades	
Flujo Básico	Paso	Acción
	1	El administrador ingresa al módulo de scan vulnerabilidades
	2	El administrador hace clic en la opción “Mostrar Resultado” del escaneo sobre el cual desea tomar acción
	3	El sistema recupera la información recolectada a nivel de base de datos
	4	El sistema muestra de forma organizada en tablas y gráficas la información referente al estado de vulnerabilidades de un dispositivo IoT
	5	Finaliza el caso de uso
Flujo Alternativo	Paso	Acción
	3.1	Si en el escaneo no se guardó información a nivel de base datos no se recolecta información
	4.1	Si en el escaneo no se guardó información a nivel de base datos no se muestra información al usuario
Postcondición	Mostrar resultado de escaneo de vulnerabilidades	

Fuente: Elaboración Propia

En la siguiente Ilustración 38 se evidencia el caso de uso para mostrar el resultado escaneo de vulnerabilidades.

Ilustración 38. Caso de Uso para Mostrar el Resultado Escaneo de Vulnerabilidades

Fuente: Elaboración Propia

3.2.4.15 Formato Caso de Uso para Crear Actividad de Remediación

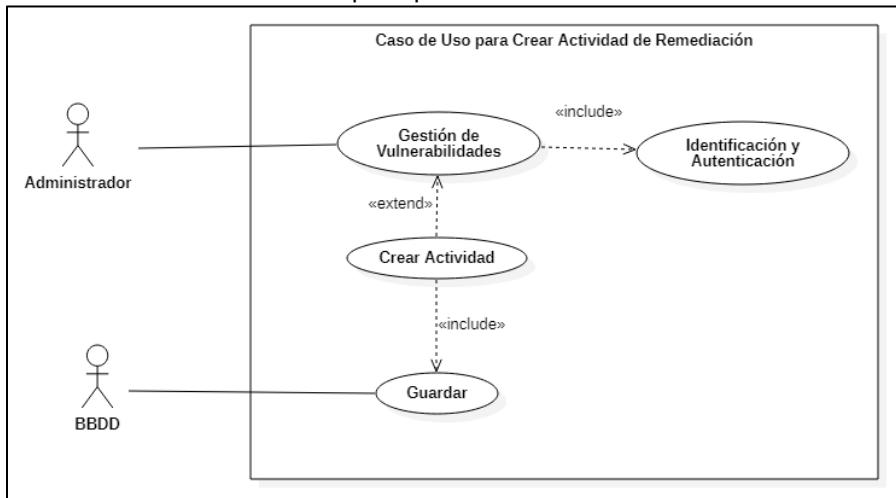
Tabla 39. Caso de Uso para Crear Actividad de Remediación

Caso de Uso	Crear Actividad de Remediación														
Actor	Administrador														
Tipo	Primario														
Descripción	Permite hacer la creación de una actividad de remediación sobre las vulnerabilidades descubiertas en un objeto IoT de un escaneo														
Precondición	El usuario debe estar autenticado y haber terminado con éxito un escaneo de vulnerabilidades														
Flujo Básico	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El administrador ingresa al módulo de gestión de vulnerabilidades</td></tr> <tr> <td>2</td><td>El administrador hace clic sobre una fecha deseada en el calendario</td></tr> <tr> <td>3</td><td>El administrador selecciona el nombre del escaneo, la vulnerabilidad, el título de la actividad de remediación, el estado, una fecha/hora inicio y una fecha/hora fin</td></tr> <tr> <td>4</td><td>El administrador hace clic en la opción “Aregar” para confirmar creación y guardar la información a nivel de base de datos</td></tr> <tr> <td>5</td><td>El sistema envía una notificación satisfactoria y agrega la actividad al calendario</td></tr> <tr> <td>6</td><td>Finaliza el caso de uso</td></tr> </tbody> </table>	Paso	Acción	1	El administrador ingresa al módulo de gestión de vulnerabilidades	2	El administrador hace clic sobre una fecha deseada en el calendario	3	El administrador selecciona el nombre del escaneo, la vulnerabilidad, el título de la actividad de remediación, el estado, una fecha/hora inicio y una fecha/hora fin	4	El administrador hace clic en la opción “Aregar” para confirmar creación y guardar la información a nivel de base de datos	5	El sistema envía una notificación satisfactoria y agrega la actividad al calendario	6	Finaliza el caso de uso
Paso	Acción														
1	El administrador ingresa al módulo de gestión de vulnerabilidades														
2	El administrador hace clic sobre una fecha deseada en el calendario														
3	El administrador selecciona el nombre del escaneo, la vulnerabilidad, el título de la actividad de remediación, el estado, una fecha/hora inicio y una fecha/hora fin														
4	El administrador hace clic en la opción “Aregar” para confirmar creación y guardar la información a nivel de base de datos														
5	El sistema envía una notificación satisfactoria y agrega la actividad al calendario														
6	Finaliza el caso de uso														
Flujo Alternativo	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>3.1</td><td>En caso que no se haya creado el escaneo de vulnerabilidades no se muestra la lista con el nombre del escaneo</td></tr> <tr> <td>3.2</td><td>En caso que no se hayan encontrado vulnerabilidades en el objeto IoT analizado no se muestran ítems el listado de vulnerabilidades</td></tr> <tr> <td>4.1</td><td>En caso que el usuario haga clic en “Cancelar” no se crea ni se guarda a nivel de base de datos el escaneo</td></tr> <tr> <td>5.1</td><td>Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no se crea el escaneo</td></tr> </tbody> </table>	Paso	Acción	3.1	En caso que no se haya creado el escaneo de vulnerabilidades no se muestra la lista con el nombre del escaneo	3.2	En caso que no se hayan encontrado vulnerabilidades en el objeto IoT analizado no se muestran ítems el listado de vulnerabilidades	4.1	En caso que el usuario haga clic en “Cancelar” no se crea ni se guarda a nivel de base de datos el escaneo	5.1	Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no se crea el escaneo				
Paso	Acción														
3.1	En caso que no se haya creado el escaneo de vulnerabilidades no se muestra la lista con el nombre del escaneo														
3.2	En caso que no se hayan encontrado vulnerabilidades en el objeto IoT analizado no se muestran ítems el listado de vulnerabilidades														
4.1	En caso que el usuario haga clic en “Cancelar” no se crea ni se guarda a nivel de base de datos el escaneo														
5.1	Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no se crea el escaneo														
Postcondición	Guarda actividad de remediación														

Fuente: Elaboración Propia

En la siguiente Ilustración 40 se evidencia el caso de uso para crear actividad de remediación.

Ilustración 39. Caso de Uso para para Crear Actividad de Remediación



Fuente: Elaboración Propia

3.2.4.16 Formato Caso de Uso para Modificar Actividad de Remediación

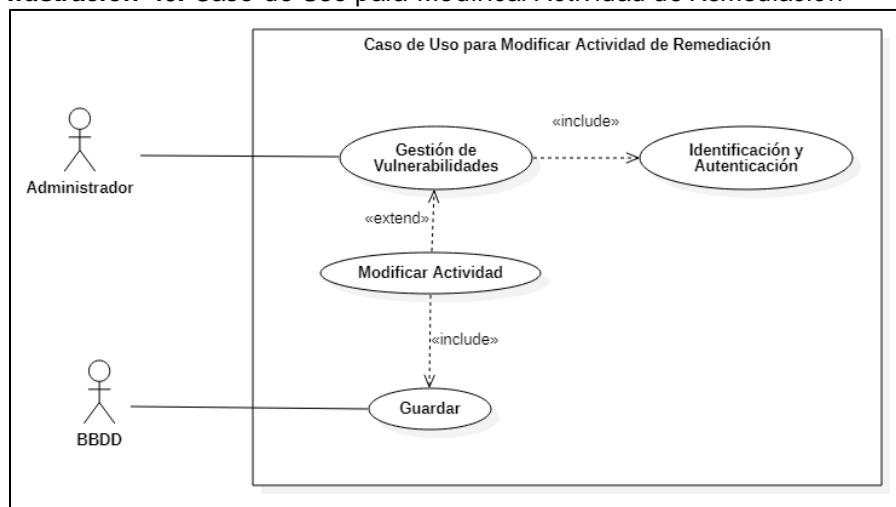
Tabla 40. Caso de Uso para Modificar Actividad de Remediación

Caso de Uso	Modificar Actividad de Remediación	
Actor	Administrador	
Tipo	Primario	
Descripción	Permite hacer la modificación de una actividad de remediación sobre las vulnerabilidades descubiertas en un objeto IoT de un escaneo	
Precondición	El usuario debe estar autenticado y debe existir una actividad de remediación previamente creada	
Flujo Básico	Paso	Acción
	1	El administrador ingresa al módulo de gestión de vulnerabilidades
	2	El administrador hace clic sobre la actividad que desea modificar
	3	El administrador puede modificar nombre de la actividad, el estado, fecha/hora inicio o fecha/hora fin
	4	El administrador hace clic en la opción "Modificar" para confirmar modificación y guardar la información a nivel de base de datos
	5	El sistema envía una notificación satisfactoria y en el calendario la información es actualizada
	6	Finaliza el caso de uso
Flujo Alternativo	Paso	Acción
	4.1	En caso que el usuario haga clic en "Cancelar" no se actualiza ni se guarda a nivel de base de datos las modificaciones de la actividad
	5.1	Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no se modifica el escaneo
Postcondición	Guarda actividad de remediación modificada	

Fuente: Elaboración Propia

En la siguiente Ilustración 41 se evidencia el caso de uso para modificar actividad de remediación.

Ilustración 40. Caso de Uso para Modificar Actividad de Remediación



Fuente: Elaboración Propia

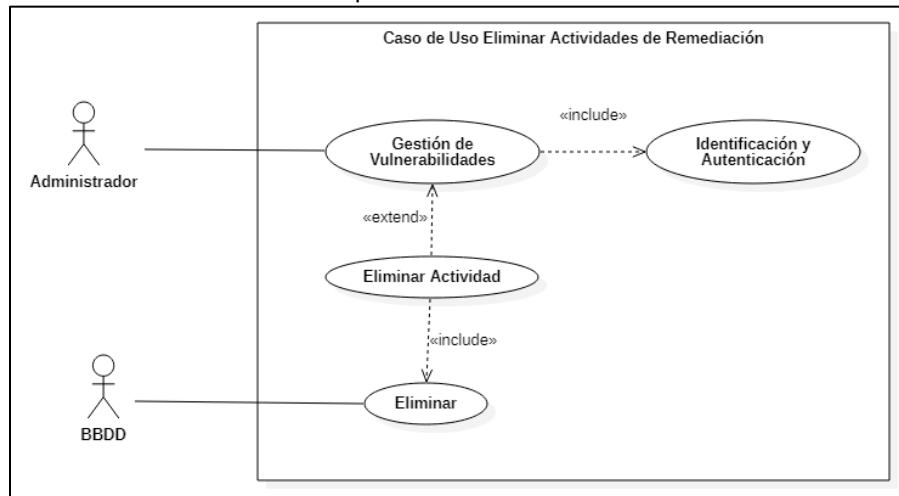
3.2.4.17 Formato Caso de Uso para Eliminar Actividades de Remediación

Tabla 41. Caso de Uso para Eliminar Actividades de Remediación

Caso de Uso	Eliminar Actividades de Remediación												
Actor	Administrador												
Tipo	Primario												
Descripción	Permite hacer la eliminación de una actividad de remediación sobre las vulnerabilidades descubiertas en un objeto IoT de un escaneo												
Precondición	El usuario debe estar autenticado y debe existir una actividad de remediación previamente creada												
Flujo Básico	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El administrador ingresa al módulo de gestión de vulnerabilidades</td></tr> <tr> <td>2</td><td>El administrador selecciona la actividad de remediación y hace clic en la opción "Borrar"</td></tr> <tr> <td>3</td><td>En el dialogo de confirmación de eliminación de actividad, el administrador hace clic en "Si, eliminar"</td></tr> <tr> <td>4</td><td>El sistema envía una notificación satisfactoria y elimina de del calendario la actividad de remediación</td></tr> <tr> <td>5</td><td>Finaliza el caso de uso</td></tr> </tbody> </table>	Paso	Acción	1	El administrador ingresa al módulo de gestión de vulnerabilidades	2	El administrador selecciona la actividad de remediación y hace clic en la opción "Borrar"	3	En el dialogo de confirmación de eliminación de actividad, el administrador hace clic en "Si, eliminar"	4	El sistema envía una notificación satisfactoria y elimina de del calendario la actividad de remediación	5	Finaliza el caso de uso
Paso	Acción												
1	El administrador ingresa al módulo de gestión de vulnerabilidades												
2	El administrador selecciona la actividad de remediación y hace clic en la opción "Borrar"												
3	En el dialogo de confirmación de eliminación de actividad, el administrador hace clic en "Si, eliminar"												
4	El sistema envía una notificación satisfactoria y elimina de del calendario la actividad de remediación												
5	Finaliza el caso de uso												
Flujo Alternativo	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>3.1</td><td>En caso que el usuario haga clic en "Cancelar" no se elimina y aún se mantiene guardado a nivel de base de datos el escaneo</td></tr> <tr> <td>4.1</td><td>Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no se elimina el escaneo</td></tr> </tbody> </table>	Paso	Acción	3.1	En caso que el usuario haga clic en "Cancelar" no se elimina y aún se mantiene guardado a nivel de base de datos el escaneo	4.1	Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no se elimina el escaneo						
Paso	Acción												
3.1	En caso que el usuario haga clic en "Cancelar" no se elimina y aún se mantiene guardado a nivel de base de datos el escaneo												
4.1	Si el sistema detecta algún tipo de anomalía o problema, este envía una notificación de error y no se elimina el escaneo												
Postcondición	Elimina actividad de remediación												

Fuente: Elaboración Propia

En la siguiente Ilustración 42 se evidencia el caso de uso para eliminar actividades de remediación.

Ilustración 41. Caso de Uso para Eliminar Actividades de Remediación

Fuente: Elaboración Propia

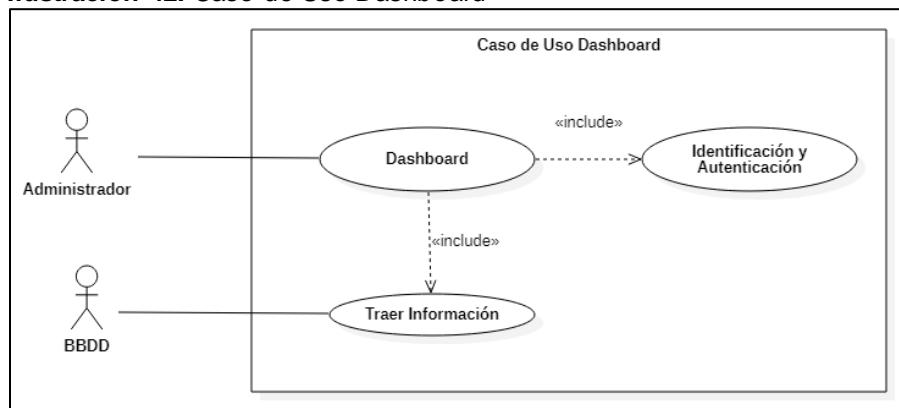
3.2.4.18 Formato Caso de Uso Dashboard

Tabla 42. Caso de Uso Dashboard

Caso de Uso	Dashboard								
Actor	Administrador								
Tipo	Primario								
Descripción	Permite de forma automática la actualización de gráficas y tablas para comprender de forma general el estado de una red a nivel de seguridad								
Precondición	El usuario debe estar autenticado								
Flujo Básico	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El administrador ingresa al módulo de dashboard</td></tr> <tr> <td>2</td><td>El sistema recolecta toda la información existente de vulnerabilidades, dispositivos, puertos, servicios y actividades de remediación, para mostrarlas en gráficas</td></tr> <tr> <td>3</td><td>Finaliza el caso de uso</td></tr> </tbody> </table>	Paso	Acción	1	El administrador ingresa al módulo de dashboard	2	El sistema recolecta toda la información existente de vulnerabilidades, dispositivos, puertos, servicios y actividades de remediación, para mostrarlas en gráficas	3	Finaliza el caso de uso
Paso	Acción								
1	El administrador ingresa al módulo de dashboard								
2	El sistema recolecta toda la información existente de vulnerabilidades, dispositivos, puertos, servicios y actividades de remediación, para mostrarlas en gráficas								
3	Finaliza el caso de uso								
Flujo Alternativo	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>2.1</td><td>Si la base de datos no cuenta con escaneos o información correspondiente a la gráfica, no carga ningún dato</td></tr> </tbody> </table>	Paso	Acción	2.1	Si la base de datos no cuenta con escaneos o información correspondiente a la gráfica, no carga ningún dato				
Paso	Acción								
2.1	Si la base de datos no cuenta con escaneos o información correspondiente a la gráfica, no carga ningún dato								
Postcondición	Carga información en gráficas de lo recolectado								

Fuente: Elaboración Propia

En la siguiente Ilustración 43 se evidencia el caso de uso dashboard.

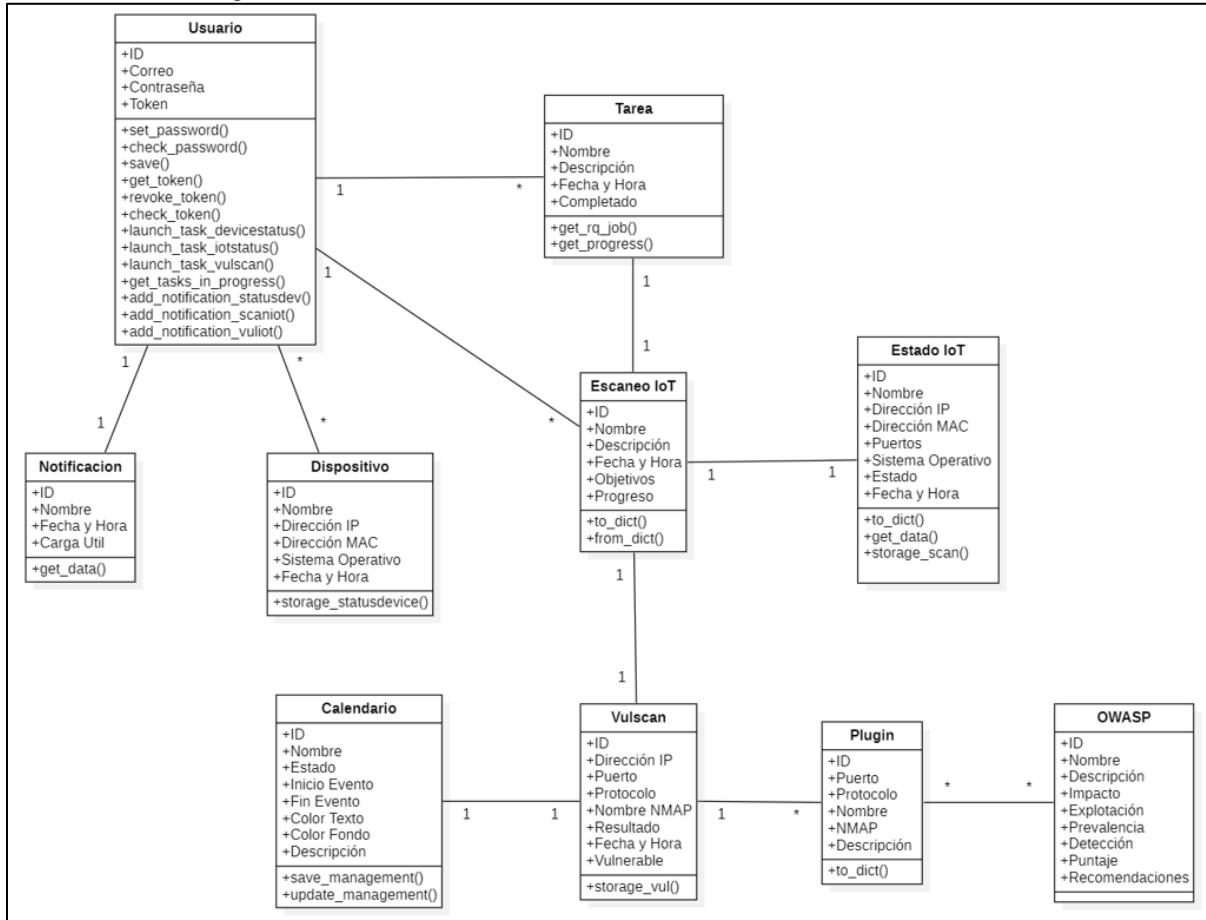
Ilustración 42. Caso de Uso Dashboard

Fuente: Elaboración Propia

3.2.5 Diagrama de Clases

Se presenta a continuación en la Ilustración 44 el diagrama de clases con el cual se adelantó todo el desarrollo e implementación de la herramienta de tal manera que se definen procesos de trabajo a nivel de desarrollo estándar.

Ilustración 43. Diagrama de Clases

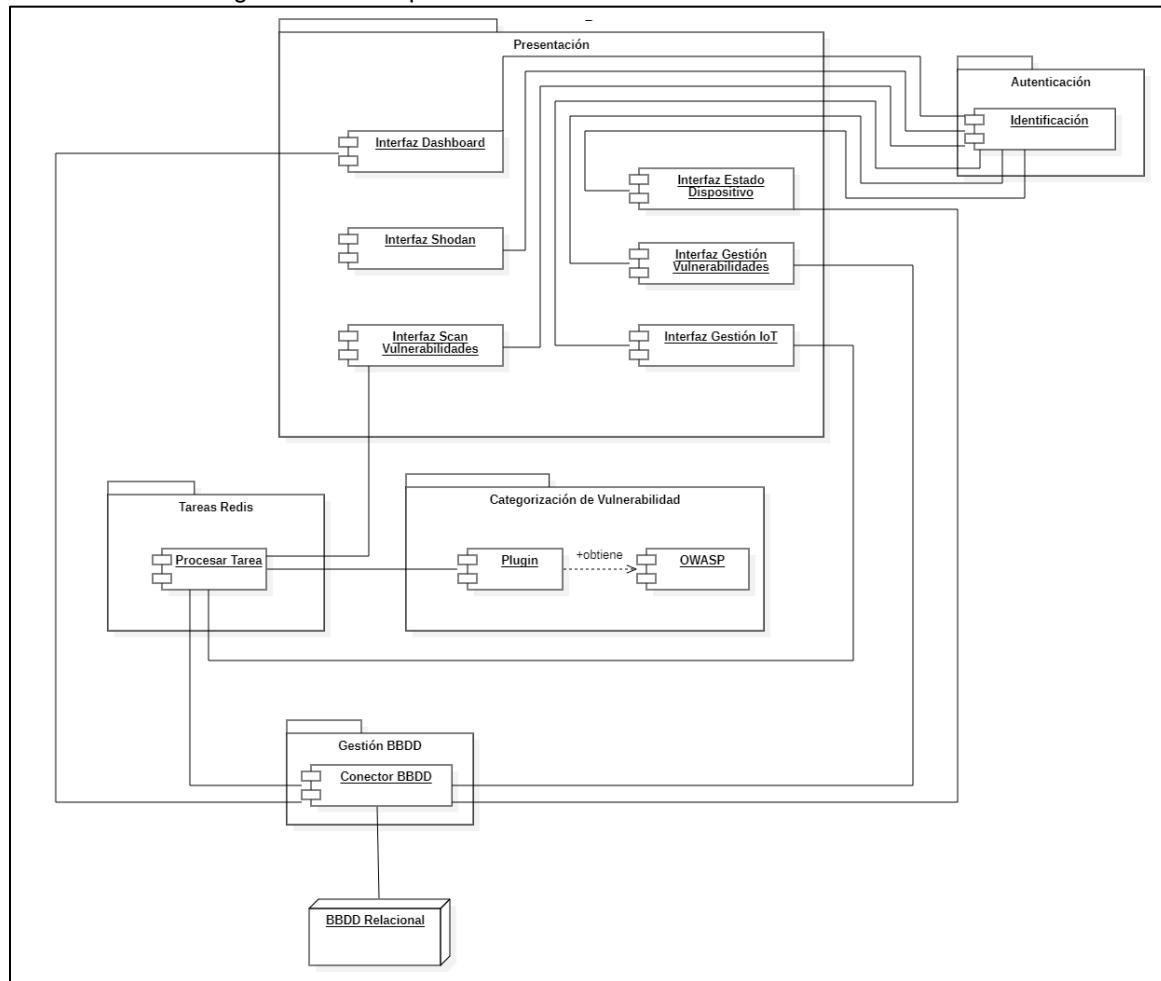


Fuente: Elaboración Propia

3.2.6 Diagrama de Componentes

Representa la forma en la que está organizada la herramienta y sus dependencias que se muestran a continuación en la Ilustración 45.

Ilustración 44. Diagrama de Componentes

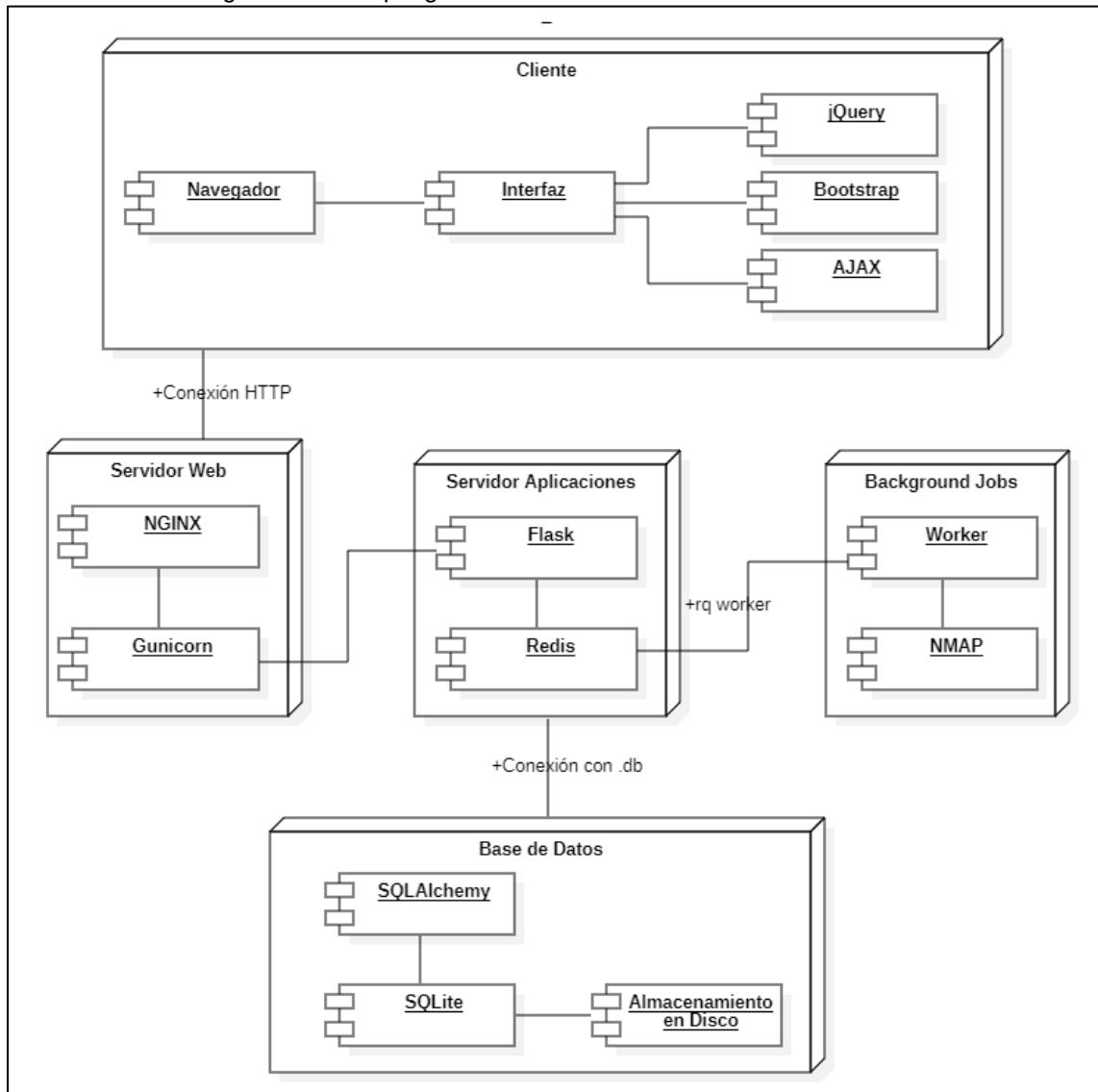


Fuente: Elaboración Propia

3.2.7 Diagrama de Despliegue

Mostrando que componentes existen y como están relacionados en la formación de interna de la herramienta, a continuación, se evidencia el diagrama de despliegue en la Ilustración 46.

Ilustración 45. Diagrama de Despliegue

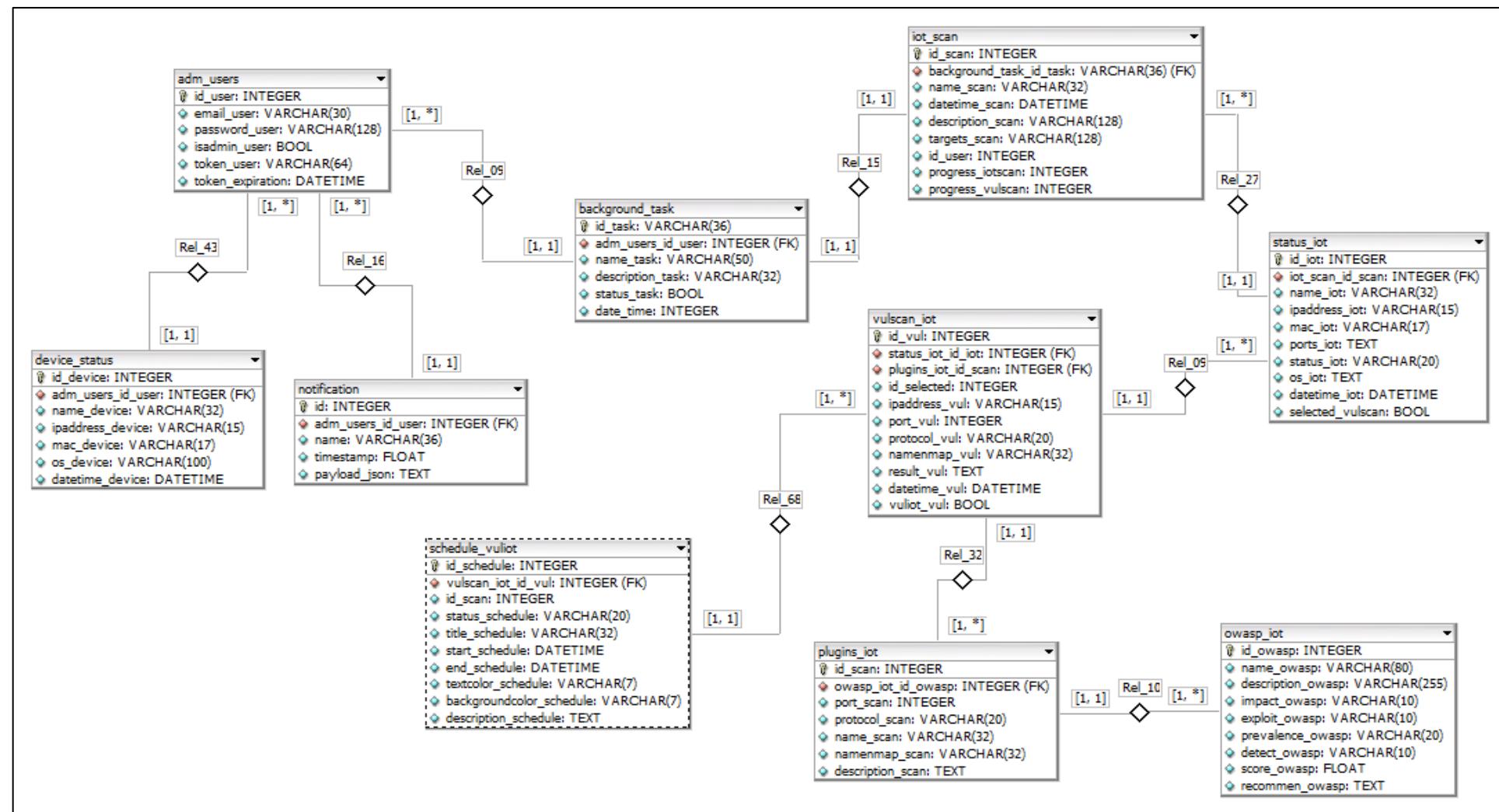


Fuente: Elaboración Propia

3.2.8 Diagrama Relacional

Para la gestión eficiente de la base de datos en la aplicación web se utiliza SQLAlchemy, que es un ORM (Object Relational Mapper), siendo importante resaltar que de igual forma existe un modelo relacional entre tablas, diseñado y definido de forma manual al momento de hacer la creación de la base de datos, por lo que la siguiente Ilustración 47 muestra el modelado de datos que permite representar las entidades relevantes de un sistema de información, así como sus interrelaciones y propiedades.

Ilustración 46. Diagrama Relacional

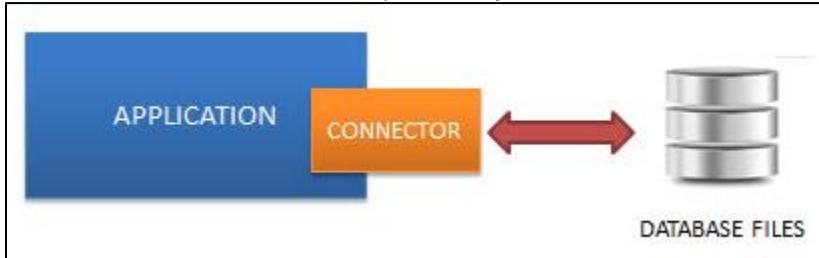


Fuente: Elaboración Propia

El motor de base de datos que se empleó para el desarrollo de la aplicación es SQLite 3, bajo el software DB Browser for SQLite de administración, al ser un archivo .bd permite la portabilidad, facilidad de uso y nivel relacional suficiente para sobrellevar de forma adecuada la carga de la aplicación. En su versión 3, SQLite permite mantener una base de datos de hasta 2 Terabytes.

Dado que es una conexión directa con la aplicación y es un archivo guardado de forma local, el nivel de latencia es casi nula y el proceso CRUD realizado para mantener el flujo de información se realiza bajo el conector de la aplicación como se muestra en la siguiente Ilustración 48:

Ilustración 47. Interacción de Aplicación y BBDD



Fuente: Tomado de <https://bit.ly/3nH9W0w>

3.2.9 Estructura API REST

Dado que el modelo de software se apoya con la biblioteca multiplataforma jQuery y la técnica de desarrollo web AJAX, se decidió hacer el uso de API REST para la transferencia cómoda de datos mientras se mantiene la comunicación asíncrona, por medio de métodos HTTP como GET, POST, PUT, etc., generando operaciones de CRUD sobre la base de datos en un formato JSON, siguiendo la estructura en diferentes acciones propias del sistema que se muestran a continuación en la Ilustración 49.

Ilustración 48. Estructura API

GET		http://192.168.0.8:2030/api/result_vul/result/1		Send	200 OK	23.7 ms	3.4 KB	Just Now
JSON	Bearer	Query	Header	Docs	Preview	Header	Cookie	Timeline
URL PREVIEW http://192.168.0.8:2030/api/result_vul/result/1								
New name New value								
<pre>{ "rows": [{ "datetime_vul": "Fri, 02 Apr 2021 09:16:59 GMT", "details_vul": "\n Supported Methods: GET HEAD POST OPTIONS", "ipaddress_vul": "192.28.115.78", "name_vul": "HTTP METHODS", "port_vul": "80/tcp", "recommen_vul": "Ninguna acci\u00f3n es requerida", "score_vul": 0.0, "summary_vul": "Informativo" }, { "datetime_vul": "Fri, 02 Apr 2021 09:16:59 GMT", "details_vul": "\n Content-type:text/html\n Connection: close\n \n (Request type: GET)\n", "ipaddress_vul": "192.28.115.78", "name_vul": "HTTP HEADER", "port_vul": "80/tcp", "recommen_vul": "Ninguna acci\u00f3n es requerida", "score_vul": 0.0, "summary_vul": "Informativo" }, { "datetime_vul": "Fri, 02 Apr 2021 09:16:59 GMT", "details_vul": "\n Directory structure:\n /\n Other:\n Longest directory structure:\n Depth: 0\n Dir: /\n Total files found (by extension):\n Other: 1\n ", "ipaddress_vul": "192.28.115.78", "name_vul": "HTTP SITEMAP", "port_vul": null }] }</pre>								

Fuente: Elaboración Propia

3.2.10 Arquitectura de Software

La arquitectura de software es una pieza fundamental de un sistema para entender sus componentes, su relación entre sí y con el medio ambiente que les rodea, por lo que se decidió hacer uso de la **arquitectura por capas** o conocida como arquitectura de tres niveles, poniendo cierto orden a la organización del código y estructura al modelo web que se estará desarrollando.

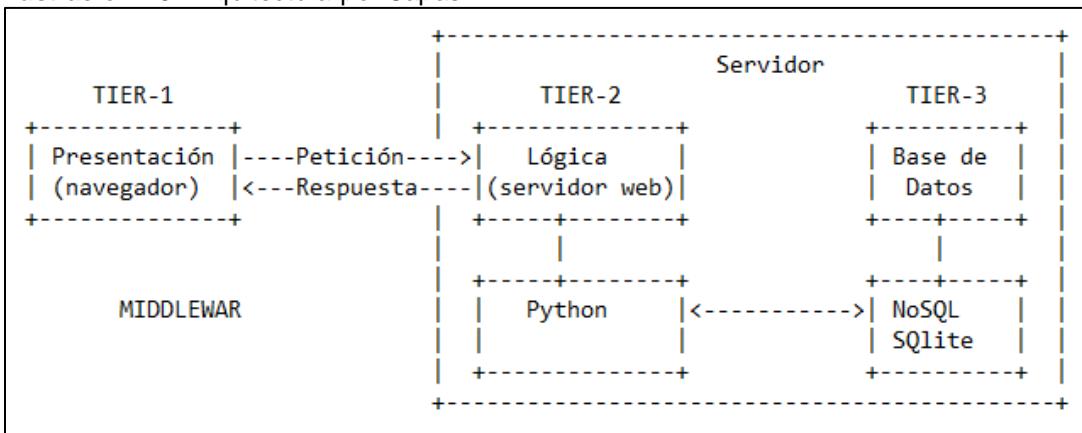
La primera capa denominada como capa de presentación o frontera, es la presentación de la interfaz gráfica ante el usuario, facilitando la interacción con la aplicación. La interfaz debe ser amigable y fácil de usar. Así mismo, en esta capa se contienen los objetos encargados de comunicar al usuario con el sistema mediante el intercambio de información, tomando y mostrando datos para hacer alguna tarea.

La segunda capa denominada como capa de lógica de negocio o control porque en esta se definen todas las reglas que se deben cumplir para la correcta ejecución del programa, así como las estructuras de datos y objetos encargados para la manipulación de los datos. Comunica con las demás capas, por ejemplo; al momento que el usuario interactúa con la capa de presentación, luego se procesa y se crean objetos de acuerdo a lo que se necesite realizar con los datos, llamado encapsulamiento, garantizando obtener la información precisa de la base de datos.

Finalmente, la tercera capa denominada de datos se encarga de realizar transacciones con la base de datos y con otros sistemas para obtener o ingresar información al sistema, valiéndose de la ya mencionada encapsulación de datos, para mantener la consistencia de los mismos (Valle, 2007).

En la siguiente Ilustración 50 se puede evidenciar la arquitectura de capas.

Ilustración 49. Arquitectura por Capas



Fuente: Elaboración Propia

3.2.11 Sprint Planning

Es la reunión que marca el inicio de cada sprint - contemplada para este proyecto que se lleve a cabo el primer lunes antes de iniciar cada sprint - participando el Product Owner, el Scrum Master y el equipo de desarrollo, se da a conocer la lista Product Backlog, correspondientes a las funciones sobre las que se van a trabajar y se establecen fechas para su cumplimiento.

3.2.11.1 Duración de Sprints

Se contempla que para el proyecto se hagan uso de nueve (9) sprints, en donde cada uno tendrá una duración de tres (3) semanas teniendo una dedicación diaria de 8 horas, dadas las especificaciones identificadas en los requerimientos y por el número de desarrolladores – correspondiente a uno (1) - que se contemplan en todo el proceso de codificación del software, quien hará labores de testing cuando sea requerido.

3.2.11.2 Valoración Historias de Usuario

Se contempla la importancia de realizar una correcta puntuación de las historias de usuario, por lo que se hace uso de la serie de Fibonacci como se observa en la Tabla 43, limitando la serie hasta el número 8 para obtener un valor en el tipo de requerimiento adecuado desde muy bajo hasta un valor muy alto en la necesidad de la tarea.

Tabla 43. Serie Fibonacci Puntuación de HU

Valor Serie Fibonacci	Estimación de Requerimiento
1	Muy Bajo
2	Bajo
3	Medio
5	Alto
8	Muy Alto

Fuente: Elaboración Propia

3.2.11.3 Planning Uno

Teniendo presente la longitud del software, se establece que se trabajara de forma simultánea entre el backend y el fronted de la aplicación web, con el fin de contar con entregables funcionales en cada sprint, así mismo dado que es un solo desarrollador quien realizará las tareas, se consideran posibles retrasos y/o variables externas que afectarán el rendimiento y cumplimiento de las fechas.

En la siguiente Ilustración 51 se muestra el cronograma del planning uno.

Ilustración 50. Cronograma Planning Uno

NOMBRE DE TAREA	TIPO DE CARACTERÍSTICA	RESPONSABLE	PUNTUACIÓN HU	INICIO	FINAL	DURACIÓN (DIAS)	ESTADO
Sprint 1		Luis Felipe N.		26/05/2020	12/06/2020	14	En progreso
Login	Seguridad	Luis Felipe N.	8	2/06/2020	5/06/2020	4	En progreso
Cambio de credenciales genéricas		Luis Felipe N.	8	9/06/2020	12/06/2020	4	En progreso
Diseño y creación de base de datos	Base de Datos	Luis Felipe N.	8	26/05/2020	28/05/2020	3	En progreso
Comunicación con base de datos		Luis Felipe N.	8	29/05/2020	29/05/2020	1	En progreso
Conexión segura a base de datos		Luis Felipe N.	8	29/05/2020	29/05/2020	1	En progreso
Credenciales seguras en base de datos		Luis Felipe N.	8	1/06/2020	1/06/2020	1	En progreso
La base de datos será implementada con trazas de auditoría		Luis Felipe N.	3	2/06/2020	2/06/2020	1	En progreso
Validación de correo	Identificación	Luis Felipe N.	5	4/06/2020	8/06/2020	5	En progreso
Validación de contraseña		Luis Felipe N.	5	4/06/2020	8/06/2020	5	En progreso
Cambio de credenciales genéricas		Luis Felipe N.	8	9/06/2020	12/06/2020	4	En progreso

Fuente: Elaboración Propia

¿Sprint Goal?

Realizar la creación adecuada de la base de datos y proporcionar un mecanismo de autenticación confiable para la aplicación web

Desarrollo de Sprint Uno

Se establecen requerimientos básicos de diseño y creación de base de datos que permitirán de forma cómoda y adecuada integrarse con los módulos que serán creados en los sprints posteriores. Así mismo la autenticación del usuario será parte fundamental al momento de llevar un control de identidad y permisos a nivel global de la aplicación web.

Sprint Backlog

El sprint backlog para este primer sprint se encuentra sujeto a lo especificado en la siguiente Tabla 44.

Tabla 44. Resumen Sprint Uno

Nombre de Proyecto	Detección de Vulnerabilidades en Dispositivos IoT
Project Manager	Luis Felipe Naranjo
Ingeniero de Desarrollo	Luis Felipe Naranjo
Número de Sprint	01
Número de Tareas	10
Día Inicio	26/05/2020
Día Final	12/06/2020
Total de Días	14

Progreso General		11,11%												
------------------	--	--------	--	--	--	--	--	--	--	--	--	--	--	--

Fuente: Elaboración Propia

En la siguiente Tabla 45 se evidencia la programación detallada del sprint uno. El burndown chart completo se encuentra en el **Anexo 8**.

Tabla 45. Programación Sprint Uno

Requisito	Tarea	Quien	Estado	Día	M	X	J	V	L	M	X	J	V	L	M	X	J	V
				Fecha	26/05/2020	27/05/2020	28/05/2020	29/05/2020	1/06/2020	2/06/2020	3/06/2020	4/06/2020	5/06/2020	8/06/2020	9/06/2020	10/06/2020	11/06/2020	12/06/2020
				Día Ejecución	1	2	3	4	5	6	7	8	9	10	11	12	13	14
				Horas Pendientes	1000	992	984	976	968	960	952	944	936	928	920	912	904	896
Sprint 1																		
Interfaz gráfica del usuario	Login	LF	En progreso							4	8	4	8	4				
	Cambio de credenciales genéricas	LF	En progreso											4	8	4	4	4
Base de datos	Diseño y creación de base de datos	LF	En progreso		8	8	8											
	Comunicación con base de datos	LF	En progreso					6										
	Conexión segura a base de datos	LF	En progreso					2										
	Credenciales seguras en base de datos	LF	En progreso						8									
	La base de datos será implementada con trazas de auditoría	LF	En progreso							4								
Autenticación y autorización	Validación de correo	LF	En progreso									2		2				
	Validación de contraseña	LF	En progreso									2		2				
	Cambio de credenciales genéricas	LF	En progreso											4		4	4	

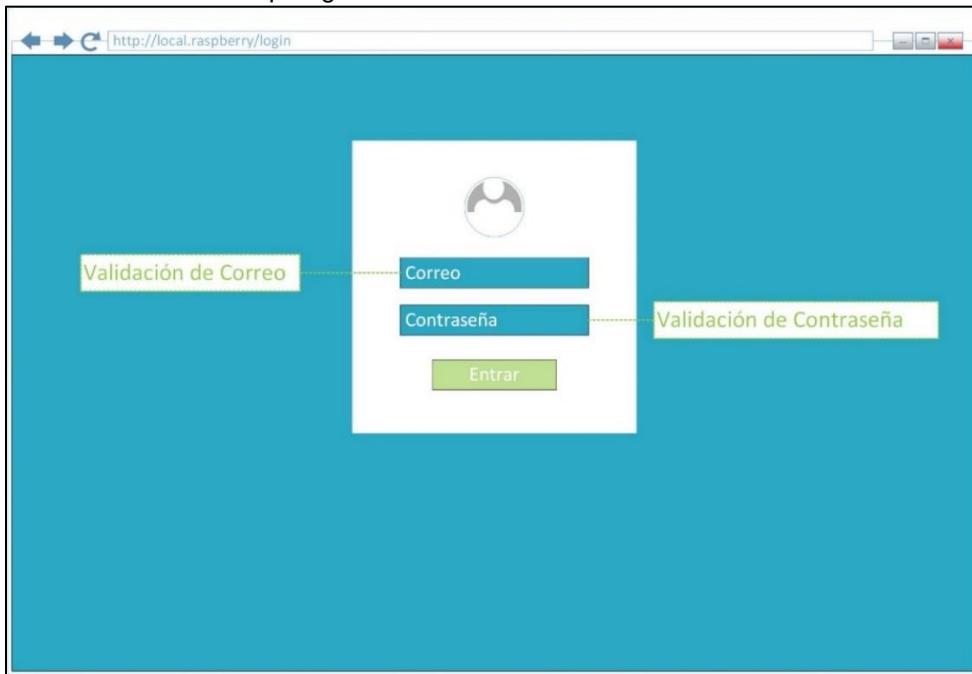
Fuente: Elaboración Propia

Mockups

Dado que se consideró desarrollar en paralelo el fronted-web de la aplicación se ha realizado la creación de mockups que ayude a entender el diseño final que se desea tener, teniendo claro que al momento final de implementación puedan contener pequeñas variaciones que ayuden a presentar una interfaz al usuario más limpia y agradable.

En la Ilustración 52 se evidencia el mockup correspondiente al login del usuario.

Ilustración 51. Mockup Login de Usuario



Fuente: Elaboración Propia

Se observa en la Ilustración 53 el mockup correspondiente al primer ingreso a la aplicación web con credenciales por defecto para su proceso de actualización.

Ilustración 52. Mockup Cambio de Credenciales por Defecto



Fuente: Elaboración Propia

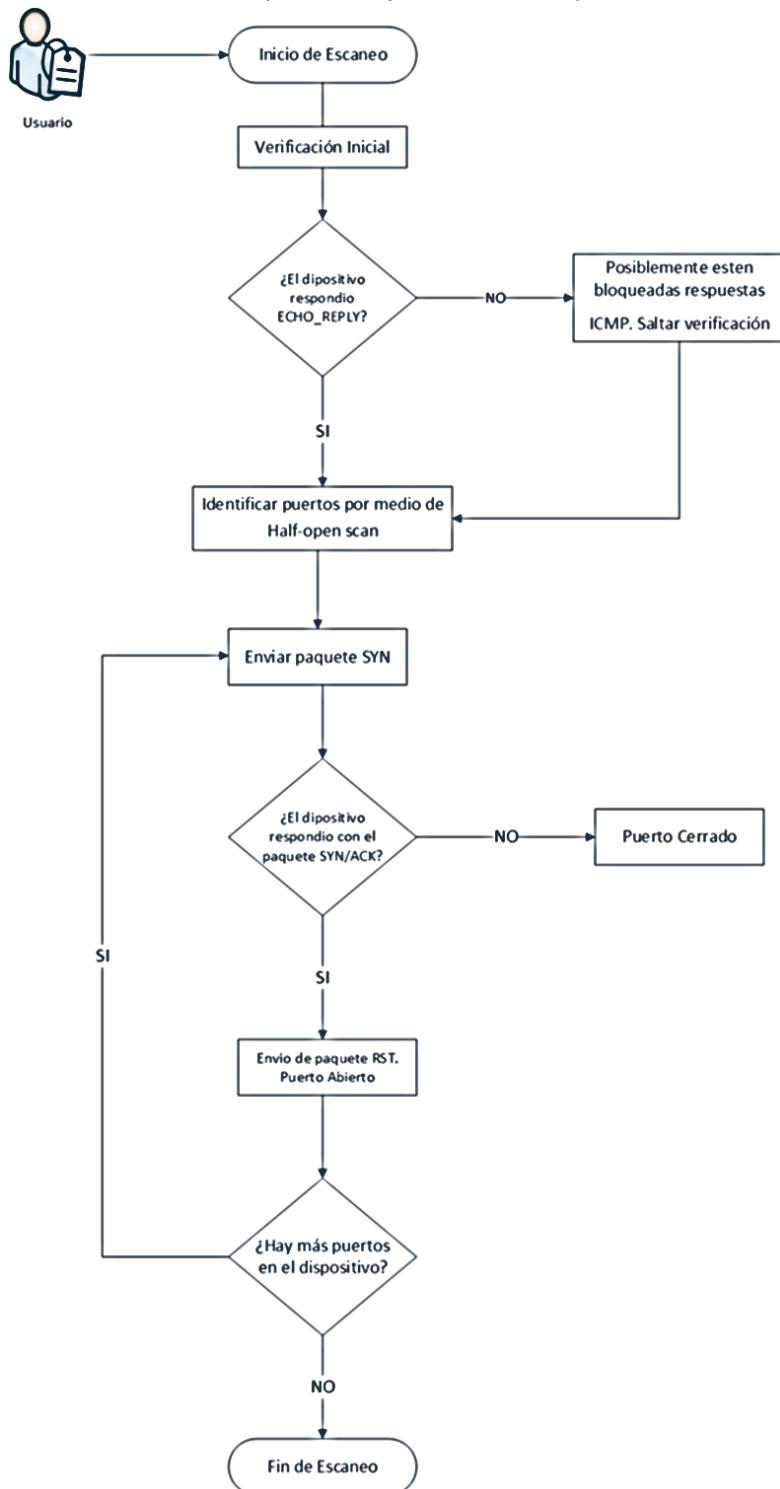
3.2.11.4 Planning Dos

Se plantea comenzar con la creación del módulo de gestión de IoT, permitiendo hacer una recolección de información a profundidad gracias a la interacción directa con el objetivo, dando la posibilidad de conocer la versión del sistema operativo, puertos abiertos, versiones de servicios publicados e información a nivel red que puede ser aprovechada por módulos posteriores para llevar a cabo la detección y explotación de vulnerabilidades de forma exitosa.

Teniendo en cuenta lo anterior, el análisis se realiza gracias a la técnica Half-open scan, que permite enviar un paquete SYN que simula establecer una conexión y espera respuesta SYN/ACK por parte del objetivo. Convirtiéndose en una técnica rápida de escaneo de dispositivos y relativamente discreta y sigilosa, que combinada con un envío nulo de paquetes ICMP puede ayudar a evitar el registro de conexiones no autorizados detectadas por IDS, IPS y firewall, lo que se convierte en una herramienta de fácil uso y con un gran potencial de recolección de información de una infraestructura.

Para realizar el proceso de descubrimiento de dispositivos, identificando información y estado, se realiza un proceso de descubrimiento que se muestra en la siguiente Ilustración 54.

Ilustración 53. Half-Open Scan y Estado de Dispositivo



Fuente: Elaboración Propia

Este proceso es pieza fundamental a hora de entender acciones internas tomadas por el software que dan garantía de recolección de datos de un objetivo y ayudan a

actividades posteriores como escaneo de vulnerabilidades que dependiendo de la información recolectada toma un proceso dinámico.

Teniendo presente la funcionalidad final a la que se quiere llegar, se establece en el planning meeting que este módulo es algo extenso, por lo que se desarrollara en el sprint segundo y el sprint tercero, así pues, se muestra a continuación en la Ilustración 55 el cronograma del planning dos.

Ilustración 54. Cronograma Planning Dos



Fuente: Elaboración Propia

¿Sprint Goal?

Realizar parte de la creación del módulo de gestión de IoT.

Retrospectiva

Finalizado satisfactoriamente el sprint uno, de forma general el proyecto y codificación de la herramienta avanza en forma adecuada acorde a los tiempos, teniendo presente que se cuenta solo con un desarrollador y factores como la falta de conocimiento a profundidad del Framework Flask de Python se han logrado sobrelevar sin percances, por lo que el uso constante y curva de aprendizaje es facilitara con el pasar el tiempo.

Ahora bien, dado que se está llevando a cabo el desarrollo en paralelo del fronted con HTML y CSS, el maqueteado del sitio web está conllevando más esfuerzo de lo esperado, por lo que a nivel técnico se recomienda aplicar de forma general el Framework Bootstrap, brindando una oportunidad de mejora, agilidad y eficiencia en el producto final.

Desarrollo de Sprint Dos

Se pretende seguir la misma dinámica aplicada al primer sprint, en donde ha sido agradable para el equipo y ha permitido evolucionar satisfactoriamente en las tareas asignadas para la ejecución de las historias de usuario. Por otra parte, se toman en cuenta las observaciones realizadas en la retrospectiva del primer sprint y serán aplicadas en este

segundo sprint, haciendo un seguimiento minucioso de la eficiencia y resultados finales que se puedan llegar a obtener para la codificación eficiente del módulo de gestión de IoT.

Sprint Backlog

El sprint backlog para este segundo sprint se encuentra sujeto a lo especificado en la siguiente Tabla 46.

Tabla 46. Resumen Sprint Dos

Nombre de Proyecto	Detección de Vulnerabilidades en Dispositivos IoT
Project Manager	Luis Felipe Naranjo
Ingeniero de Desarrollo	Luis Felipe Naranjo
Número de Sprint	02
Número de Tareas	10
Día Inicio	16/06/2020
Día Final	3/07/2020
Total de Días	12
Progreso General	22,22%

Fuente: Elaboración Propia

En la siguiente Tabla 47 se evidencia la programación detallada del sprint dos.

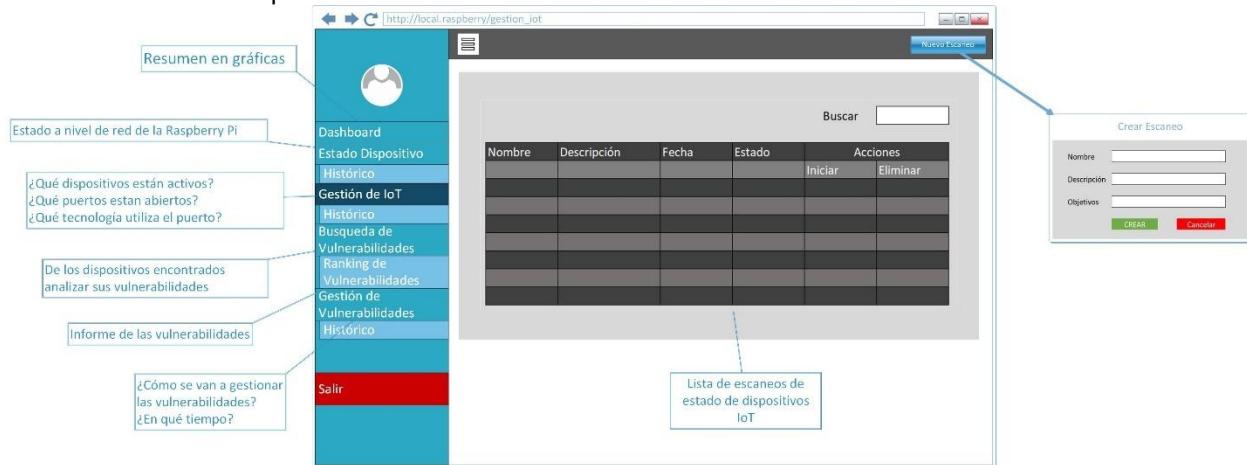
Tabla 47. Programación Sprint Dos

Requisito	Tarea	Quien	Estado	Día	M	X	J	V	M	X	J	V	M	X	J	V
				Fecha	16/06/2020	17/06/2020	18/06/2020	19/06/2020	23/06/2020	24/06/2020	25/06/2020	26/06/2020	30/06/2020	1/07/2020	2/07/2020	3/07/2020
				Día Ejecución	15	16	17	18	19	20	21	22	23	24	25	26
				Horas Pendientes	888	880	872	864	856	848	840	832	824	816	808	800
Sprint 2																
Interfaz gráfica del usuario	Crear escaneo	LF	En progreso		8	6	2									
	Modificar escaneo	LF	En progreso							4						
	Eliminar escaneo	LF	En progreso								4	2				
	Barra de progreso escaneo	LF	En progreso										3			
	Iniciar escaneo	LF	En progreso												1	
Modulo gestión de IoT	Crear escaneo	LF	En progreso			2	6	8								
	Modificar escaneo	LF	En progreso						8	4						
	Eliminar escaneo	LF	En progreso								4	6				
	Barra de progreso escaneo	LF	En progreso									5				
	Iniciar escaneo	LF	En progreso										8	7	8	

Fuente: Elaboración Propia

Mockups

Se establece de forma tentativa el uso de la siguiente Ilustración 56 como interfaz de usuario para la gestión de IoT y navegabilidad de la herramienta ubicada en la parte izquierda.

Ilustración 55. Mockup Módulo de Gestión de IoT

Fuente: Elaboración Propia

3.2.11.5 Planning Tres

Se planea continuar y terminar el módulo de gestión de IoT en este sprint tercero (3), tomando los parámetros y prácticas empleadas a lo largo del desarrollo que han tenido resultados favorables, agilidad en el proceso final y un producto adecuado a los requerimientos necesitados.

En la siguiente Ilustración 57 se muestra el cronograma del planning tres.

Ilustración 56. Cronograma Planning Tres

NOMBRE DE TAREA	TIPO DE CARACTERÍSTICA	RESPONSABLE	PUNTUACIÓN HU	INICIO	FINAL	DURACIÓN (DIAS)	ESTADO
Sprint 3				6/07/2020	24/07/2020	14	En progreso
Resultado escaneo	Funcionalidad	Luis Felipe N.	5	6/07/2020	10/07/2020	5	En progreso
Organizar y clasificar en cada campo el parámetro encontrado en los dispositivos IoT		Luis Felipe N.	5	13/07/2020	14/07/2020	2	En progreso
El sistema controlará el acceso a escaneos propios y lo permitirá solamente a usuarios autorizados		Luis Felipe N.	8	15/07/2020	15/07/2020	1	En progreso
Tarjetas estado escaneos	Seguimiento	Luis Felipe N.	2	16/07/2020	17/07/2020	2	En progreso
Copiar tabla escaneo		Luis Felipe N.	2	21/07/2020	21/07/2020	1	En progreso
Filtro escaneo		Luis Felipe N.	2	22/07/2020	22/07/2020	1	En progreso
Exportar tabla escaneos		Luis Felipe N.	2	23/07/2020	23/07/2020	1	En progreso
Exportar resultado escaneo		Luis Felipe N.	2	23/07/2020	23/07/2020	1	En progreso
Buscar dispositivo en resultado de escaneo		Luis Felipe N.	2	24/07/2020	24/07/2020	1	En progreso

23/06/2020 28/06/2020 3/07/2020 8/07/2020 13/07/2020 18/07/2020 23/07/2020 28/07/2020

sprint 3

Resultado escaneo

Organizar y clasificar en cada campo el parámetro encontrado en los dispositivos IoT

El sistema controlará el acceso a escaneos propios y lo permitirá solamente a usuarios autorizados

Tarjetas estado escaneos

Copiar tabla escaneo

Filtro escaneo

Exportar tabla escaneos

Exportar resultado escaneo

Buscar dispositivo en resultado de escaneo

Fuente: Elaboración Propia

¿Sprint Goal?

Terminar la creación del módulo de gestión de IoT.

Retrospectiva

Finalizado satisfactoriamente el sprint dos, se ha mantenido el ritmo y el modelo de trabajo de forma óptima, en donde los integrantes del equipo se han sentido agustos en la codificación del software, encontrando formas de optimizarlo y que sea integrado de forma fácil con los demás módulos.

Para que sea más interactivo, dinámico y sea actualizado la información de forma inmediata, se recomienda hacer uso de la extensión DataTables y Bootstrap Tables que integran jQuery, facilitando llevar una estandarización de forma global que ayude a agilizar el proceso de construcción de la herramienta.

Desarrollo de Sprint Tres

La carga establecida para ejecución de las tareas de los sprint ha sido optima y se han podido cumplir de forma satisfactoria, fortaleciendo la célula de trabajo y cada vez se tiene más confianza en el uso de las herramientas.

A lo recomendado, se hará uso de las extensiones DataTables y Bootstrap Tables para organizar los escaneos creados, modificados, ejecución y presentación final de los resultados obtenidos, así mismo se plantea hacer uso Redis integrado en Python para crear "hilos" que ayuden con la ejecución asíncrona de tareas tanto para el escaneo de estado de objetos IoT como para el módulo de escaneo de vulnerabilidades, dando la facilidad de uso al usuario final para navegar por la aplicación web sin que sea truncado un escaneo y poder tener un control en la barra de progreso.

Sprint Backlog

El sprint backlog para este tercer sprint se encuentra sujeto a lo especificado en la siguiente Tabla 48.

Tabla 48. Resumen Sprint Tres

Nombre de Proyecto	Detección de Vulnerabilidades en Dispositivos IoT
Project Manager	Luis Felipe Naranjo
Ingeniero de Desarrollo	Luis Felipe Naranjo
Número de Sprint	03
Número de Tareas	17
Día Inicio	6/07/2020
Día Final	24/07/2020
Total de Días	14
Progreso General	33,33%

Fuente: Elaboración Propia

En la siguiente Tabla 49 se evidencia la programación detallada del sprint tres.

Tabla 49. Programación Sprint Tres

Requisito	Tarea	Quien	Estado	Fecha	6/07/2020	7/07/2020	8/07/2020	9/07/2020	10/07/2020	11/07/2020	14/07/2020	15/07/2020	16/07/2020	17/07/2020	21/07/2020	22/07/2020	23/07/2020	24/07/2020	
					Día Ejecución	27	28	29	30	31	32	33	34	35	36	37	38	39	40
					Horas Pendientes	792	784	776	768	760	752	744	736	728	720	712	704	696	688
Sprint 3																			
Interfaz gráfica del usuario	Resultado escaneo	LF	En progreso		6	4			8										
	Organizar y clasificar en cada campo el parámetro encontrado en los dispositivos IoT	LF	En progreso							5									
	Tarjetas estado escaneos	LF	En progreso										6						
	Copiar tabla escaneo	LF	En progreso											1					
	Filtro escaneo	LF	En progreso											2					
	Exportar tabla escaneos	LF	En progreso											1					
	Exportar resultado escaneo	LF	En progreso											1					
Modulo gestión de IoT	Buscar dispositivo en resultado de escaneo	LF	En progreso															1	
	Resultado escaneo	LF	En progreso		2	4	8	8											
	Organizar y clasificar en cada campo el parámetro encontrado en los dispositivos IoT	LF	En progreso							3	8								
	El sistema controlará el acceso a escaneos propios y lo permitirá solamente a usuarios autorizados	LF	En progreso									8							
	Tarjetas estado escaneos	LF	En progreso										2	8					
	Copiar tabla escaneo	LF	En progreso											7					
	Filtro escaneo	LF	En progreso											6					

Fuente: Elaboración Propia

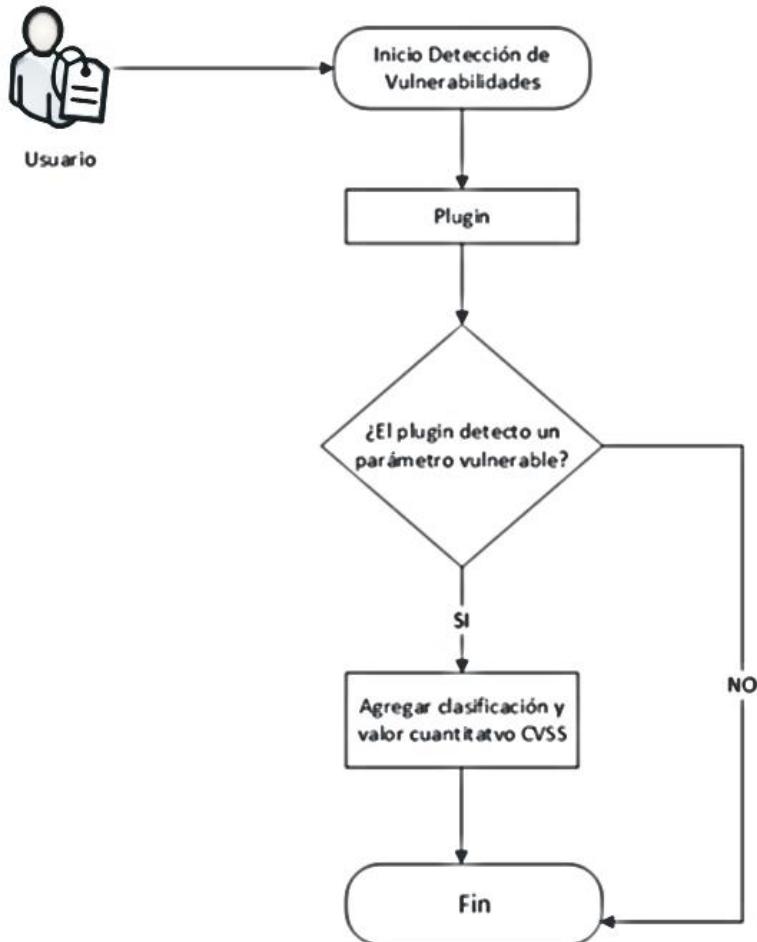
3.2.11.6 Planning Cuatro

Se plantea comenzar con la creación del módulo scan vulnerabilidades, en donde recolectada la información del objeto IoT en el módulo anterior, esta es usada como ayuda para detectar vulnerabilidades en objetos IoT. Al seleccionar el objeto IoT y los plugins – complementos del software que permiten hacer la comprobación y explotación de alguna vulnerabilidad – se muestran apartados de resultados a lo encontrado, posicionando en tres (3) diferentes rankings compuestos por:

- **Vulnerabilidades:** Ranking principal posicionado según la severidad encontrada en el o los dispositivos, en donde se muestra un resumen de la vulnerabilidad, resultado de explotación de la vulnerabilidad y recomendaciones necesarias para su posible remediación.
- **IoT:** Dependiendo de la cantidad de objetivos que se hayan deseado escanear, sus vulnerabilidades se muestran en un ranking, en donde se posiciona y se muestra que tan vulnerable es un dispositivo y de acuerdo con los plugins se hace un conteo de clasificación según sea una vulnerabilidad crítica, alta, media, baja o informativa.
- **Puertos:** Se genera un ranking de acuerdo con la severidad de las vulnerabilidades encontrados en los puertos del objetivo o los objetivos, dando la posibilidad a un administrador de identificar cuales puertos se encuentran más propensos a ser atacados por agentes internos o externos de una organización.

Para entender de forma clara el proceso que se pretende desarrollar en este módulo se muestra en la siguiente Ilustración 58 el flujo adecuado de ejecución.

Ilustración 57. Escaneo de Vulnerabilidades en Objetos IoT



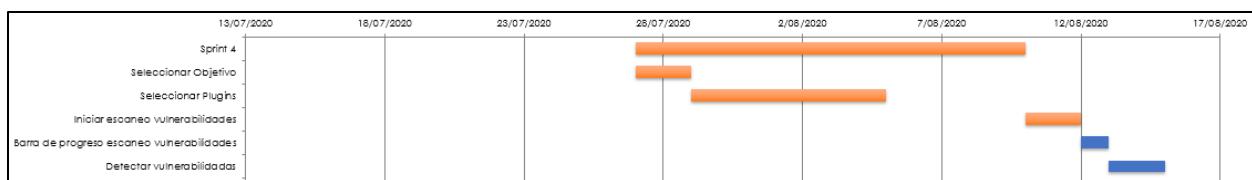
Fuente: Elaboración Propia

Este proceso es la esencia de la aplicación web y el fundamento del proyecto, por lo que entender acciones internas deseadas tomadas por el software, dan garantía de un resultado acorde a lo necesario para el escaneo de vulnerabilidades sobre objetos IoT.

Se establece en el planning meeting que este módulo al igual que el anterior es algo extenso, por lo que se desarrollara en el sprint cuarto, quinto y sexto, así pues, se muestra a continuación en la Ilustración 59 el cronograma del planning cuatro.

Ilustración 58. Cronograma Planning Cuatro

NOMBRE DE TAREA	TIPO DE CARACTERÍSTICA	RESPONSABLE	PUNTUACIÓN HU	INICIO	FINAL	DURACIÓN (DIAS)	ESTADO
Sprint 4				27/07/2020	14/08/2020	14	En progreso
Seleccionar Objetivo	Funcionalidad	Luis Felipe N.	5	27/07/2020	28/07/2020	2	En progreso
Seleccionar Plugins		Luis Felipe N.	5	29/07/2020	6/08/2020	7	En progreso
Iniciar escaneo vulnerabilidades		Luis Felipe N.	5	10/08/2020	11/08/2020	2	En progreso
Barra de progreso escaneo vulnerabilidades		Luis Felipe N.	5	12/08/2020	12/08/2020	1	En progreso
Detectar vulnerabilidades	Necesidad de Negocio	Luis Felipe N.	8	13/08/2020	14/08/2020	2	En progreso



Fuente: Elaboración Propia

¿Sprint Goal?

Realizar parte de la creación del módulo de escaneo de vulnerabilidades.

Retrospectiva

Finalizado satisfactoriamente el sprint tres, se presentaron problemas en la integración de las extensiones DataTables y Bootstrap Tables, dado que el API REST creado en Python no poseía la estructura para el funcionamiento con estos plugins de jQuery JavaScript, por lo que fue necesario re-estructurarlo y ajustarlo, lo cual consumió más esfuerzo en la ejecución de este sprint para poder cumplir con los tiempos. Así mismo, la integración de Redis para la ejecución recurrentes asíncronas de tareas tuvo inconvenientes debido a la falta de conocimientos en profundidad y poca documentación encontrada.

De forma general se cumplió satisfactoriamente con el tiempo y las actividades, sin embargo, se recomienda la no integración adicional de frameworks o librerías complejas que puedan dar como consecuencia retrasos en las entregas y conlleven esfuerzos adicionales no previstos.

Desarrollo de Sprint Cuatro

Se pretende seguir la misma dinámica aplicada a lo largo de los sprint ejecutados, sin embargo, el nivel de complejidad y asignación de tareas van a ser más bajas, en compensación al anterior sprint tres, además dando la posibilidad de un tiempo prudente de auto-aprendizaje a nivel técnico y personal que ayuden entender como mejorar el proceso de codificación para obtener resultados óptimos.

Sprint Backlog

El sprint backlog para este segundo sprint se encuentra sujeto a lo especificado en la siguiente Tabla 50.

Tabla 50. Resumen Sprint Cuatro

Nombre de Proyecto	Detección de Vulnerabilidades en Dispositivos IoT
Project Manager	Luis Felipe Naranjo
Ingeniero de Desarrollo	Luis Felipe Naranjo
Número de Sprint	04
Número de Tareas	09
Día Inicio	27/07/2020
Día Final	14/08/2020
Total de Días	14

Progreso General		44,44%
Fuente: Elaboración Propia		

En la siguiente Tabla 51 se evidencia la programación detallada del sprint cuatro.

Tabla 51. Programación Sprint Cuatro

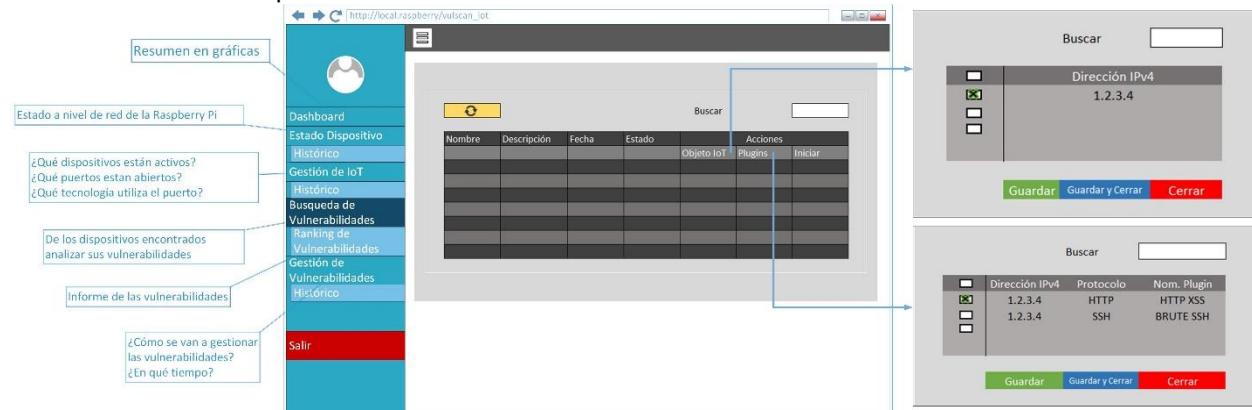
Requisito	Tarea	Quien	Estado	Día	L	M	X	J	V	S	M	X	J	L	M	X	J	V
				Fecha	27/07/2020	28/07/2020	29/07/2020	30/07/2020	31/07/2020	3/08/2020	4/08/2020	5/08/2020	6/08/2020	10/08/2020	11/08/2020	12/08/2020	13/08/2020	14/08/2020
				Ejecución	41	42	43	44	45	46	47	48	49	50	51	52	53	54
				Horas Pendientes	680	672	664	656	648	640	632	624	616	608	600	592	584	576
Sprint 4																		
Interfaz gráfica del usuario	Seleccionar Objetivo	LF	En progreso		6	4												
	Seleccionar Plugins	LF	En progreso							4	5	2	3					
	Iniciar escaneo vulnerabilidades	LF	En progreso														1	
	Barra de progreso escaneo vulnerabilidades	LF	En progreso															3
Modulo escaneo de vulnerabilidades	Seleccionar Objetivo	LF	En progreso		2	4												
	Seleccionar Plugins	LF	En progreso					8	8	4	3	6	8	5				
	Iniciar escaneo vulnerabilidades	LF	En progreso														7	5
	Barra de progreso escaneo vulnerabilidades	LF	En progreso															8
	Detectar vulnerabilidades	LF	En progreso															8

Fuente: Elaboración Propia

Mockups

Se establece de forma tentativa el uso de la siguiente Ilustración 60 como interfaz de usuario para el escaneo de vulnerabilidades sobre objetos IoT.

Ilustración 59. Mockup Módulo Scan de Vulnerabilidades



Fuente: Elaboración Propria

3.2.11.7 Planning Cinco

Se planea continuar el módulo de scan vulnerabilidades comenzado en este sprint cuarto (4), teniendo presente los parámetros y prácticas empleadas a lo largo del desarrollo que han tenido resultados favorables, agilidad en el proceso final y un producto adecuado a los requerimientos necesitados.

En la siguiente Ilustración 61 se muestra el cronograma del planning cinco

Ilustración 60. Cronograma Planning Cinco

NOMBRE DE TAREA	TIPO DE CARACTERÍSTICA	RESPONSABLE	PUNTUACIÓN HU	INICIO	FINAL	DURACIÓN (DIAS)	ESTADO
Sprint 5				18/08/2020	4/09/2020	14	En progreso
Resultado escaneo vulnerabilidades	Evaluación de Resultados	Luis Felipe N.	5	18/08/2020	18/08/2020	1	En progreso
Lista de hallazgos escaneo de vulnerabilidades		Luis Felipe N.	5	19/08/2020	21/08/2020	3	En progreso
Identificar y clasificar la criticidad de vulnerabilidades		Luis Felipe N.	8	24/08/2020	25/08/2020	2	En progreso
Clasificar remedición de vulnerabilidades	Remedición	Luis Felipe N.	8	25/08/2020	27/08/2020	3	En progreso
Mostrar resultado de explotación	Vulnerabilidad	Luis Felipe N.	5	28/08/2020	28/08/2020	1	En progreso
Evaluar a nivel de seguridad el impacto para dispositivo IoT		Luis Felipe N.	8	28/08/2020	1/09/2020	3	En progreso
Por cada dispositivo IoT Identificar y clasificar las vulnerabilidades encontradas	Clasificación de Vulnerabilidades	Luis Felipe N.	8	1/09/2020	3/09/2020	3	En progreso
Evaluar nivel de impacto a nivel de seguridad para cada puerto descubierto		Luis Felipe N.	8	3/09/2020	4/09/2020	2	En progreso

The Gantt chart illustrates the timeline for Sprint 5 tasks. The x-axis represents dates from 7/08/2020 to 11/09/2020. The y-axis lists tasks: Sprint 5, Resultado escaneo vulnerabilidades, Lista de hallazgos escaneo de vulnerabilidades, Identificar y clasificar la criticidad de vulnerabilidades, Clasificar remedición de vulnerabilidades, Mostrar resultado de explotación, Evaluar a nivel de seguridad el impacto para dispositivo IoT, Por cada dispositivo IoT Identificar y clasificar las vulnerabilidades encontradas, and Evaluar nivel de impacto a nivel de seguridad para cada puerto descubierto. Task durations are color-coded: Resultado escaneo (orange), Lista de hallazgos (orange), Identificar y clasificar (orange), Clasificar remedición (blue), Mostrar resultado (blue), Evaluar a nivel de seguridad (blue), Por cada dispositivo IoT (blue), and Evaluar nivel de impacto (green).

Fuente: Elaboración Propia

¿Sprint Goal?

Continuar con las funcionalidades necesarias para cumplir con lo necesitado en el módulo de scan de vulnerabilidades.

Retrospectiva

Finalizado satisfactoriamente el sprint cuarto, la carga asignada disminuyó bastante, por lo que el equipo logró tomar un tiempo adicional para avanzar en la curva de aprendizaje de los Frameworks y librerías necesitadas para desarrollo exitoso del proyecto, así pues, la dinámica adoptada fue óptima para retomar esfuerzos y lograr un dinamismo que agilice el proceso de trabajo que se está abordando, el cual se pretende continuar para mejorar la calidad del producto final.

Desarrollo de Sprint Cinco

El núcleo del software se encuentra en el desarrollo del módulo de scan de vulnerabilidades, por lo que se pretende fomentar en el equipo el compromiso necesario para desarrollar y tener el nivel de análisis/comprendión necesaria para poder clasificar de forma adecuada las vulnerabilidades, moldeando un esquema propio bajo algunas metodologías del mercado que ayuden a evidenciar el nivel de exposición de amenazas del objeto IoT.

La información es el pilar de todo proceso de auditoría y análisis en toma de decisiones y por ello la fórmula debe constar de varias consideraciones a saber para que sea implementado en el software como:

- Determinar estado de vulnerabilidad

- Dados unos parámetros internos del sistema, se determina de acuerdo con un análisis y reconocimiento del resultado si el sistema es vulnerable a cierto tipo de amenaza.
- Calificación definida por cada ítem de evaluación (Completo, organización, claridad).
- Tipo de vulnerabilidad
 - Con base en el proyecto OWASP Top 10 IoT y el IoT Security Compliance Framework. Siendo importante resaltar que de las 10 consideraciones propuestas por OWASP Internet of Things en su versión actual 2018 se tomaron 4 parámetros implementados en el software desarrollado y ayudan a categorizar las vulnerabilidades encontradas en un dispositivo IoT:
 - Contraseñas débiles, adivinables o codificadas
 - Servicios de red inseguros
 - Interfaces inseguras del ecosistema
 - Protección de privacidad insuficiente

Esta categorización determina en primera instancia determinar que fallo presenta un dispositivo y que procesos de remediación se adapta mejor a la vulnerabilidad.

El proceso completo de clasificación se puede observar en el Anexo 4.

- Calificación de vulnerabilidad
 - Teniendo como base la métrica de evaluación de vulnerabilidades denominada como CVSS (Common Vulnerability Scoring System) en su versión 3.0 se clasifica el impacto y severidad de las vulnerabilidades encontradas en un sistema.

Sprint Backlog

El sprint backlog para este quinto sprint se encuentra sujeto a lo especificado en la siguiente Tabla 52.

Tabla 52. Resumen Sprint Cinco

Nombre de Proyecto	Detección de Vulnerabilidades en Dispositivos IoT
Project Manager	Luis Felipe Naranjo
Ingeniero de Desarrollo	Luis Felipe Naranjo
Número de Sprint	05
Equipo de Desarrollo	01
Número de Tareas	16
Día Inicio	18/08/2020
Día Final	4/09/2020
Total de Días	14
Progreso General	55,55%

Fuente: Elaboración Propia

En la siguiente Tabla 53 se evidencia la programación detallada del sprint cinco.

Tabla 53. Programación Sprint Cinco

Requisito	Tarea	Quien	Estado	Dia	M	X	J	Y	L	M	X	J	Y	L	M	X	J	Y
				Fecha	18/08/2020	19/08/2020	20/08/2020	21/08/2020	24/08/2020	25/08/2020	26/08/2020	27/08/2020	28/08/2020	31/08/2020	1/09/2020	2/09/2020	3/09/2020	4/09/2020
				Dia Ejecución	55	56	57	58	59	60	61	62	63	64	65	66	67	68
				Horas Pendientes	568	560	552	544	536	528	520	512	504	496	488	480	472	464
Sprint 5																		
Interfaz gráfica del usuario	Resultado escaneo vulnerabilidades	LF	En progreso		2													
	Lista de hallazgos escaneo de	LF	En progreso			7		4										
	Identificar y clasificar la criticidad de vulnerabilidades	LF	En progreso						2	2								
	Clasificar remediación de vulnerabilidades	LF	En progreso									4	5					
	Mostrar resultado de explotación	LF	En progreso										3					
	Evaluar a nivel de seguridad el impacto para dispositivo IoT	LF	En progreso										5	2				
	Por cada dispositivo IoT Identificar y clasificar las vulnerabilidades encontradas	LF	En progreso											3	2			
	Evaluar nivel de impacto a nivel de seguridad para cada puerto descubierto	LF	En progreso															3
Modulo escaneo de vulnerabilidades	Resultado escaneo vulnerabilidades	LF	En progreso		6													
	Lista de hallazgos escaneo de	LF	En progreso			1	8	4										
	Identificar y clasificar la criticidad de vulnerabilidades	LF	En progreso						6	5								
	Clasificar remediación de vulnerabilidades	LF	En progreso							1	4	3						
	Mostrar resultado de explotación	LF	En progreso									3						
	Evaluar a nivel de seguridad el impacto para dispositivo IoT	LF	En progreso									2	3	4				
	Por cada dispositivo IoT Identificar y clasificar las vulnerabilidades encontradas	LF	En progreso										2	5	5			
	Evaluar nivel de impacto a nivel de seguridad para cada puerto descubierto	LF	En progreso														1	5

Fuente: Elaboración Propia

Mockups

Se establece de forma tentativa el uso de la siguiente Ilustración 62 como interfaz de usuario para mostrar el resultado de vulnerabilidades sobre objetos IoT.

Ilustración 61. Mockup Resultado de Vulnerabilidades



Fuente: Elaboración Propia

3.2.11.8 Planning Seis

Se planea finalizar el módulo de scan vulnerabilidades comenzado en este sprint cuarto (4), agregando funcionalidad de apoyo al resultado de escaneo de vulnerabilidades para que facilite la visualización de la información y navegabilidad final sobre las vulnerabilidades detectadas. Se pretende continuar haciendo uso de las técnicas empleadas a lo largo del desarrollo que han tenido resultados favorables y con la calidad que es esperada.

En la siguiente Ilustración 63 se muestra el cronograma del planning seis.

Ilustración 62. Cronograma Planning Seis

NOMBRE DE TAREA	TIPO DE CARACTERÍSTICA	RESPONSABLE	PUNTUACIÓN HU	INICIO	FINAL	DURACIÓN (DIAS)	ESTADO
Sprint 6				7/09/2020	25/09/2020	15	En progreso
Gráfica resultado escaneo de vulnerabilidades	Gráfica resumen	Luis Felipe N.	3	7/09/2020	9/09/2020	3	En progreso
Apartado resumen de escaneo vulnerabilidades		Luis Felipe N.	3	10/09/2020	10/09/2020	1	En progreso
Expandir/Contraer ítems resultado escaneo vulnerabilidades		Luis Felipe N.	3	11/09/2020	14/09/2020	2	En progreso
Tarjetas estado escaneos de vulnerabilidades	Visibilidad de estados	Luis Felipe N.	2	14/09/2020	15/09/2020	2	En progreso
Copiar tabla escaneo vulnerabilidades		Luis Felipe N.	2	16/09/2020	16/09/2020	1	En progreso
Filtro escaneo vulnerabilidades		Luis Felipe N.	2	16/09/2020	16/09/2020	1	En progreso
Exportar tabla escaneos vulnerabilidades	Funcionalidad	Luis Felipe N.	2	17/09/2020	17/09/2020	1	En progreso
Copiar tabla resultado escaneo de vulnerabilidades		Luis Felipe N.	2	18/09/2020	18/09/2020	1	En progreso
Filtro tabla resultado escaneo de vulnerabilidades		Luis Felipe N.	2	21/09/2020	22/09/2020	2	En progreso
Exportar tabla resultado escaneo vulnerabilidades		Luis Felipe N.	2	22/09/2020	22/09/2020	1	En progreso
Buscar vulnerabilidad en resultado escaneo vulnerabilidades		Luis Felipe N.	2	23/09/2020	23/09/2020	1	En progreso
Buscar dispositivo en resultado de escaneo vulnerabilidades		Luis Felipe N.	2	24/09/2020	24/09/2020	1	En progreso
Buscar puerto en resultado de escaneo vulnerabilidades		Luis Felipe N.	2	25/09/2020	25/09/2020	1	En progreso

Fuente: Elaboración Propia

¿Sprint Goal?

Finalizar con las funcionalidades necesarias para cumplir con lo necesario en el módulo de scan de vulnerabilidades.

Retrospectiva

Finalizado satisfactoriamente el sprint cinco, se ha mantenido un ritmo de trabajo constante y en este punto del proyecto el equipo trabaja de forma adecuada, mostrando un nivel de compromiso acorde a las necesidades del proyecto y obteniendo un resultado de vulnerabilidades estructurado a lo solicitado.

Se ha logrado disminuir gracias a la planeación conveniente de los sprint en gran medida impases que pueden retrasar la entrega del producto final, así pues, solo es recomendable empezar con la búsqueda de una librería en JavaScript que permita implementar un calendario de forma ágil para el módulo de gestión de vulnerabilidad, aventajando cuestiones de aprendizaje necesario y evitar cualquier tipo de percance.

Desarrollo de Sprint Seis

Para este sprint se plantea fortalecer el producto y mejorar aquellos aspectos deficientes en el desarrollo del módulo de scan de vulnerabilidades en sprint anteriores, de forma general se aplican funcionalidades que pueden ser abordadas por el Framework de Data Tables por lo que sin importar el gran número de tareas asignadas pueden ser solucionadas de forma rápida y dar pie a una revisión general del módulo para poder ser optimizado, garantizando un resultado que supere las expectativas.

Sprint Backlog

El sprint backlog para este sexto sprint se encuentra sujeto a lo especificado en la siguiente Tabla 54.

Tabla 54. Resumen Sprint Seis

Nombre de Proyecto	Detección de Vulnerabilidades en Dispositivos IoT
Project Manager	Luis Felipe Naranjo
Ingeniero de Desarrollo	Luis Felipe Naranjo
Número de Sprint	06
Número de Tareas	26
Día Inicio	7/09/2020
Día Final	25/09/2020
Total de Días	15
Progreso General	66,66%

Fuente: Elaboración Propia

En la siguiente Tabla 55 se evidencia la programación detallada del sprint seis

Tabla 55. Programación Sprint Seis

Requisito	Tarea	Quien	Estado	Dia	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V
				Fecha	7/09/2020	8/09/2020	9/09/2020	10/09/2020	11/09/2020	14/09/2020	15/09/2020	16/09/2020	17/09/2020	18/09/2020	21/09/2020	22/09/2020	23/09/2020	24/09/2020	25/09/2020
				Dia Ejecución	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83
				Horas Pendientes	456	448	440	432	424	416	408	400	392	384	376	368	360	352	344
Sprint 6																			
Interfaz gráfica del usuario	Gráfica resultado escaneo de vulnerabilidades	LF	En progreso			6	5	2											
	Apartado resumen de escaneo vulnerabilidades	LF	En progreso						5										
	Expandir/Contraer items resultado escaneo vulnerabilidades	LF	En progreso							8									
	Tarjetas estado escaneos de vulnerabilidades	LF	En progreso								4	2							
	Copiar tabla escaneo vulnerabilidades	LF	En progreso									2							
	Filtro escaneo vulnerabilidades	LF	En progreso									2							
	Exportar tabla escaneos vulnerabilidades	LF	En progreso									3							
	Copiar tabla resultado escaneo de vulnerabilidades	LF	En progreso										2						
	Filtro tabla resultado escaneo de vulnerabilidades	LF	En progreso										3						
	Exportar tabla resultado escaneo vulnerabilidades	LF	En progreso											2					
	Buscar vulnerabilidad en resultado escaneo vulnerabilidades	LF	En progreso											3					
	Buscar dispositivo en resultado de escaneo vulnerabilidades	LF	En progreso											3					
	Buscar puerta en resultado de escaneo vulnerabilidades	LF	En progreso												3				
Modulo escaneo de vulnerabilidades	Gráfica resultado escaneo de vulnerabilidades	LF	En progreso		2	3	6												
	Apartado resumen de escaneo vulnerabilidades	LF	En progreso					3											
	Expandir/Contraer items resultado escaneo vulnerabilidades	LF	En progreso						3										
	Tarjetas estado escaneos de vulnerabilidades	LF	En progreso							1	6								
	Copiar tabla escaneo vulnerabilidades	LF	En progreso								2								
	Filtro escaneo vulnerabilidades	LF	En progreso								2								
	Exportar tabla escaneos vulnerabilidades	LF	En progreso								5								
	Copiar tabla resultado escaneo de vulnerabilidades	LF	En progreso									6							
	Filtro tabla resultado escaneo de vulnerabilidades	LF	En progreso										5	3					
	Exportar tabla resultado escaneo vulnerabilidades	LF	En progreso										3						
	Buscar vulnerabilidad en resultado escaneo vulnerabilidades	LF	En progreso											5					
	Buscar dispositivo en resultado de escaneo vulnerabilidades	LF	En progreso											5					
	Buscar puerta en resultado de escaneo vulnerabilidades	LF	En progreso												5				

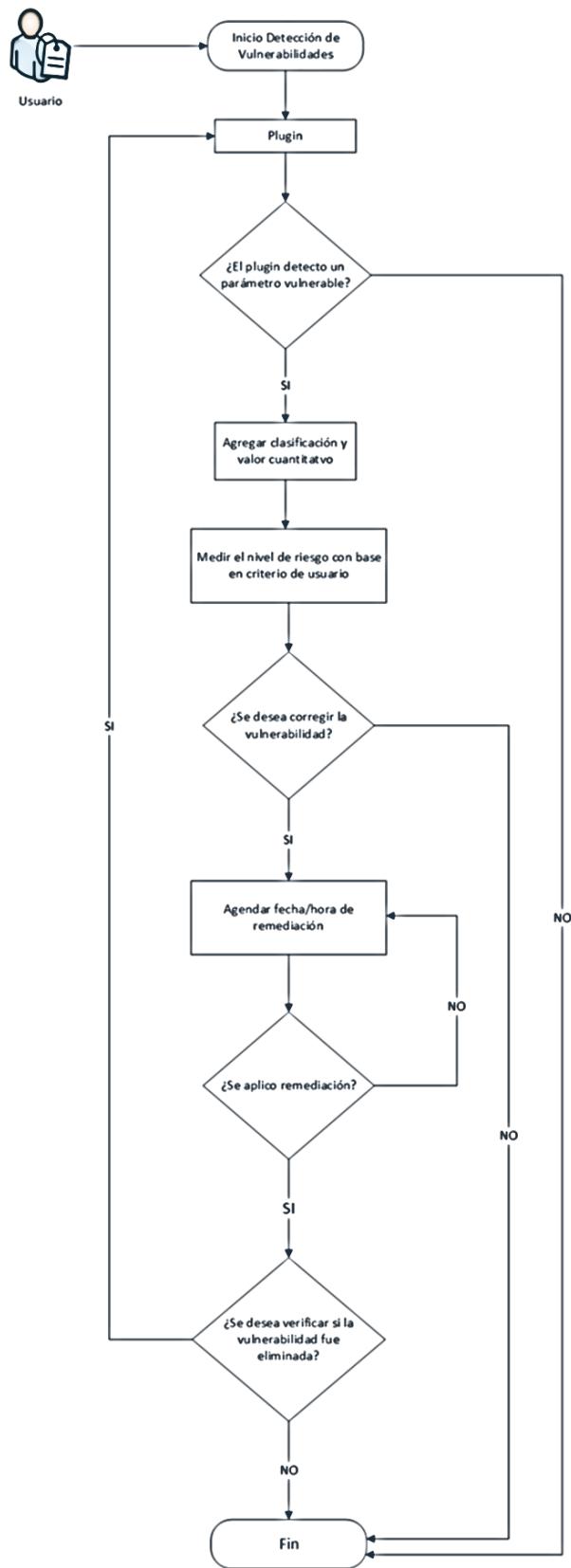
Fuente: Elaboración Propia

3.2.11.9 Planning Siete

Para este sprint se tiene como meta hacer la creación del módulo de gestión de vulnerabilidades, en el cual mediante un calendario es posible agendar una fecha/hora en la cual el equipo responsable de la remediación de este tipo vulnerabilidades se encargue de solucionarlas, especificando el estado en el que se encuentra esta tarea de remediación, entre las cuales se encuentran; creado, pendiente ejecución, pausado, cancelado o solucionado, brindando un espacio en donde se mantenga un control adecuado de las vulnerabilidades encontradas, garantizando que sean solucionadas y evitando no solo su detección sino un proceso de solución óptimo y organizado entre equipos de trabajo.

Para realizar el proceso descrito, es necesario valerse de módulos desarrollados en anteriores sprints, por lo que se trata de hacer un seguimiento y control al ciclo de vida de una vulnerabilidad que se puede evidenciar en la siguiente Ilustración 64.

Ilustración 63. Gestión de Ciclo de Vida de Vulnerabilidad



Fuente: Elaboración Propia

En la siguiente Ilustración 65 se muestra el cronograma del planning siete.

Ilustración 64. Cronograma Planning Siete

NOMBRE DE TAREA	TIPO DE CARACTERÍSTICA	RESPONSABLE	PUNTUACIÓN HU	INICIO	FINAL	DURACIÓN (DIAS)	ESTADO
Sprint 7				28/09/2020	16/10/2020	14	En progreso
Calendario principal	Funcionalidad	Luis Felipe N.	5	28/09/2020	29/09/2020	2	En progreso
Crear gestión de vulnerabilidad	Seguimiento	Luis Felipe N.	5	30/09/2020	5/10/2020	4	En progreso
Mostrar vulnerabilidades detectadas para su gestión		Luis Felipe N.	5	6/10/2020	7/10/2020	2	En progreso
Modificar gestión de vulnerabilidades	Gestión	Luis Felipe N.	5	8/10/2020	13/10/2020	3	En progreso
Borrar gestión de vulnerabilidades		Luis Felipe N.	5	14/10/2020	15/10/2020	2	En progreso
Filtros de tiempo calendario		Luis Felipe N.	3	16/10/2020	16/10/2020	1	En progreso

The Gantt chart illustrates the timeline for Sprint 7 tasks. The x-axis represents dates from 16/09/2020 to 21/10/2020. The y-axis lists tasks: Sprint 7, Calendario principal, Crear gestión de vulnerabilidad, Mostrar vulnerabilidades detectadas para su gestión, Modificar gestión de vulnerabilidades, Borrar gestión de vulnerabilidades, and Filtros de tiempo calendario. Task durations are color-coded: orange for 'Crear gestión de vulnerabilidad' (4 days), orange for 'Mostrar vulnerabilidades detectadas para su gestión' (2 days), blue for 'Modificar gestión de vulnerabilidades' (3 days), blue for 'Borrar gestión de vulnerabilidades' (2 days), and blue for 'Filtros de tiempo calendario' (1 day). The 'Calendario principal' task is listed but has no visible bar.

Fuente: Elaboración Propia

¿Sprint Goal?

Realizar la creación de las funcionalidades necesarias para el módulo de gestión de vulnerabilidades.

Retrospectiva

Finalizado satisfactoriamente el sprint sexto, se lograron abordar gran cantidad de tareas asociadas a la historia de usuario para mostrar resultados en un escaneo de vulnerabilidades sobre un objeto IoT, gracias a la facilidad de los frameworks y librerías utilizadas, el tiempo de ejecución del sprint se logró hacer en la mitad de lo estipulado, dando la posibilidad de hacer la actividad tentativa para una revisión general al módulo, alcanzando a mejorar aspectos de funcionalidad, rendimiento, estructuración de los API's REST y mejorar en la interfaz gráfica del usuario.

A esta altura del proyecto no se prevé hacer algún tipo de modificación a la estrategia de trabajo empleada, por lo que de forma general se seguirá apoyando al equipo en las herramientas y/o percances que se presenten, para continuar ofreciendo una calidad adecuada del producto final.

Desarrollo de Sprint Siete

Contemplando las necesidades identificadas en el sprint planning meeting de intentar realizar un seguimiento del ciclo de vida de una vulnerabilidad que en este caso abordaría la tecnología IoT, por lo que en la creación de este módulo se procura seguir lineamientos generales del mercado, para que sean implementados en la codificación del software y permitan tener una remediación adecuada y eficiente de las vulnerabilidades encontradas.

Sprint Backlog

El sprint backlog para este séptimo sprint se encuentra sujeto a lo especificado en la siguiente Tabla 56.

Tabla 56. Resumen Sprint Siete

Nombre de Proyecto	Detección de Vulnerabilidades en Dispositivos IoT
Project Manager	Luis Felipe Naranjo
Ingeniero de Desarrollo	Luis Felipe Naranjo
Número de Sprint	07
Número de Tareas	12
Día Inicio	28/09/2020
Día Final	16/10/2020
Total de Días	14
Progreso General	77,77%

Fuente: Elaboración Propia

En la siguiente Tabla 57 se evidencia la programación detallada del sprint siete.

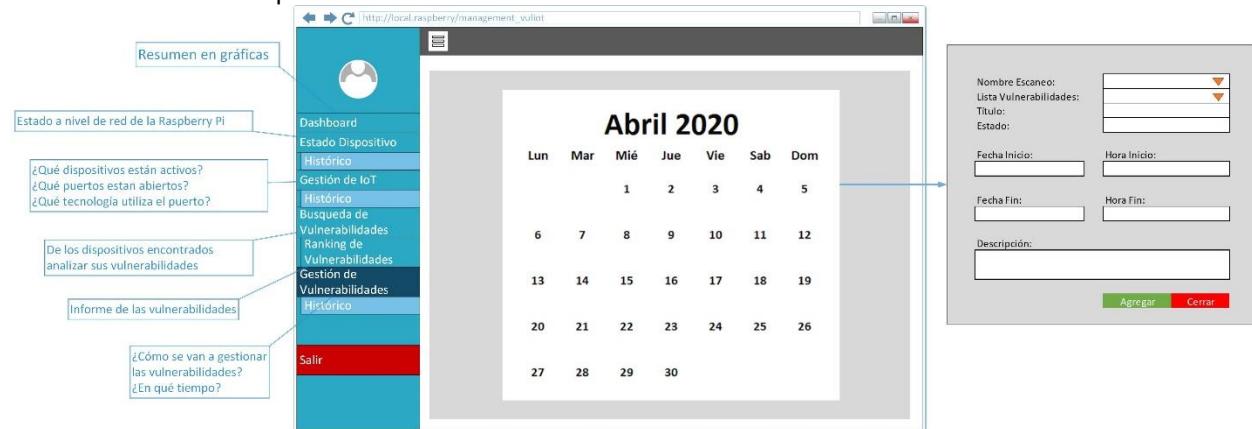
Tabla 57. Programación Sprint Siete

Requisito	Tarea	Quien	Estado	Día	L	M	X	J	V	L	M	X	J	V	M	X	J	V
				Fecha	28/09/2020	29/09/2020	30/09/2020	1/10/2020	2/10/2020	5/10/2020	6/10/2020	7/10/2020	8/10/2020	9/10/2020	13/10/2020	14/10/2020	15/10/2020	16/10/2020
				Día Ejecución	84	85	86	87	88	89	90	91	92	93	94	95	96	97
				Horas Pendientes	336	328	320	312	304	296	288	280	272	264	256	248	240	232
Sprint 7																		
Interfaz gráfica del usuario	Calendario principal	LF	En progreso		6	2			4	6	5		2					
	Crear gestión de vulnerabilidad	LF	En progreso									1	4					
	Mostrar vulnerabilidades detectadas para su gestión	LF	En progreso											4	2	3		
	Modificar gestión de vulnerabilidades	LF	En progreso													6	1	
	Borrar gestión de vulnerabilidades	LF	En progreso															2
Modulo gestión de vulnerabilidades	Filtros de tiempo calendario	LF	En progreso															
	Calendario principal	LF	En progreso		2	6												
	Crear gestión de vulnerabilidad	LF	En progreso				8	4	2	3		2						
	Mostrar vulnerabilidades detectadas para su gestión	En progreso										7		4				
	Modificar gestión de vulnerabilidades	LF	En progreso											6	5			
	Borrar gestión de vulnerabilidades	LF	En progreso												2	7		
	Filtros de tiempo calendario	LF	En progreso															6

Fuente: Elaboración Propia

Mockups

Se establece de forma tentativa el uso de la siguiente Ilustración 66 como interfaz de usuario para el módulo de gestión de vulnerabilidades.

Ilustración 65. Mockup Gestión de Vulnerabilidades

Fuente: Elaboración Propia

3.2.11.10 Planning Ocho

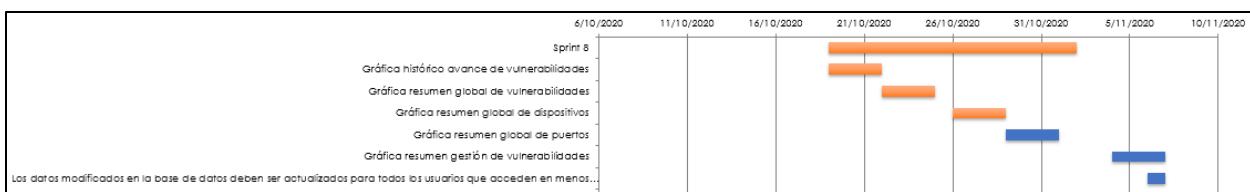
Avanzando favorablemente en el desarrollo del producto, se pretende brindar la posibilidad de mostrar de forma gráfica el resumen global, por lo que para este sprint se tiene como meta hacer la creación del módulo de dashboard, visualizando cinco (5) gráficos de forma resumida del conglomerado de resultados obtenidos de los diferentes módulos que comprende al software, en donde se encuentran:

1. Conteo general de vulnerabilidades encontradas por mes en dispositivos.
2. Resumen global de vulnerabilidades encontradas en dispositivos escaneados, clasificados acorde a su severidad – informativo, bajo, medio, alto o critico – conociendo de forma general que tan vulnerable se puede encontrar una red.
3. Criticidad de las vulnerabilidades encontradas en los diferentes dispositivos, clasificados por ranking - direcciones IP – con base en el riesgo de exposición frente amenazas que represente el dispositivo.
4. Criticidad de las vulnerabilidades encontradas en los puertos de los diferentes dispositivos escaneados con base en el riesgo de exposición frente amenazas que represente cada uno.
5. Resumen de agenda por mes de las acciones tomadas por el personal encargado de mitigar las vulnerabilidades en una organización.

En la siguiente Ilustración 67 se muestra el cronograma del planning ocho.

Ilustración 66. Cronograma Planning Ocho

NOMBRE DE TAREA	TIPO DE CARACTERÍSTICA	RESPONSABLE	PUNTUACIÓN HU	INICIO	FINAL	DURACIÓN (DIAS)	ESTADO
Sprint 8				19/10/2020	6/11/2020	14	En progreso
Gráfica histórico avance de vulnerabilidades	Gráficas	Luis Felipe N.	3	19/10/2020	21/10/2020	3	En progreso
Gráfica resumen global de vulnerabilidades		Luis Felipe N.	3	22/10/2020	26/10/2020	3	En progreso
Gráfica resumen global de dispositivos		Luis Felipe N.	3	26/10/2020	28/10/2020	3	En progreso
Gráfica resumen global de puertos		Luis Felipe N.	3	29/10/2020	3/11/2020	3	En progreso
Gráfica resumen gestión de vulnerabilidades		Luis Felipe N.	3	4/11/2020	6/11/2020	3	En progreso
Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.	Rendimiento	Luis Felipe N.	3	6/11/2020	6/11/2020	1	En progreso



Fuente: Elaboración Propia

¿Sprint Goal?

Realizar la creación con las funcionalidades necesarias para el módulo de gestión de vulnerabilidades.

Retrospectiva

Finalizado satisfactoriamente el sprint séptimo, el nivel de comprensión y agilidad en el desarrollo del software ha permitido avanzar normalmente por las diferentes series de tareas asignadas al equipo, por lo que la disciplina aplicada muestra en estos momentos resultados palpables que se pueden demostrar en un avance de producto de calidad y acorde a las necesidades del cliente.

Desarrollo de Sprint Ocho

El trabajo realizado a lo largo de los sprint ha sido favorable en la construcción de la proactividad del equipo, abordando soluciones excelentes al momento de presentarse percances en el desarrollo del proyecto, reflejado en la reunión de retrospectiva, en donde procesos deficientes fueron descartados y se cuentas con feedback positivos de forma general, fortaleciendo la estructura del equipo.

En este sprint se prevé la creación del módulo principal del dashboard acorde a las especificaciones y tareas asignadas en el sprint planning meeting, por lo que las expectativas son altas, faltando tan solo 1 sprint para culminar de forma total el proyecto.

Sprint Backlog

El sprint backlog para este octavo sprint se encuentra sujeto a lo especificado en la siguiente Tabla 58.

Tabla 58. Resumen Sprint Ocho

Nombre de Proyecto	Detección de Vulnerabilidades en Dispositivos IoT
Project Manager	Luis Felipe Naranjo
Ingeniero de Desarrollo	Luis Felipe Naranjo
Número de Sprint	08
Número de Tareas	11
Día Inicio	19/10/2020
Día Final	6/11/2020
Total de Días	14
Progreso General	88,88%

Fuente: Elaboración Propia

En la siguiente Tabla 59 se evidencia la programación detallada del sprint ocho

Tabla 59. Programación Sprint Ocho

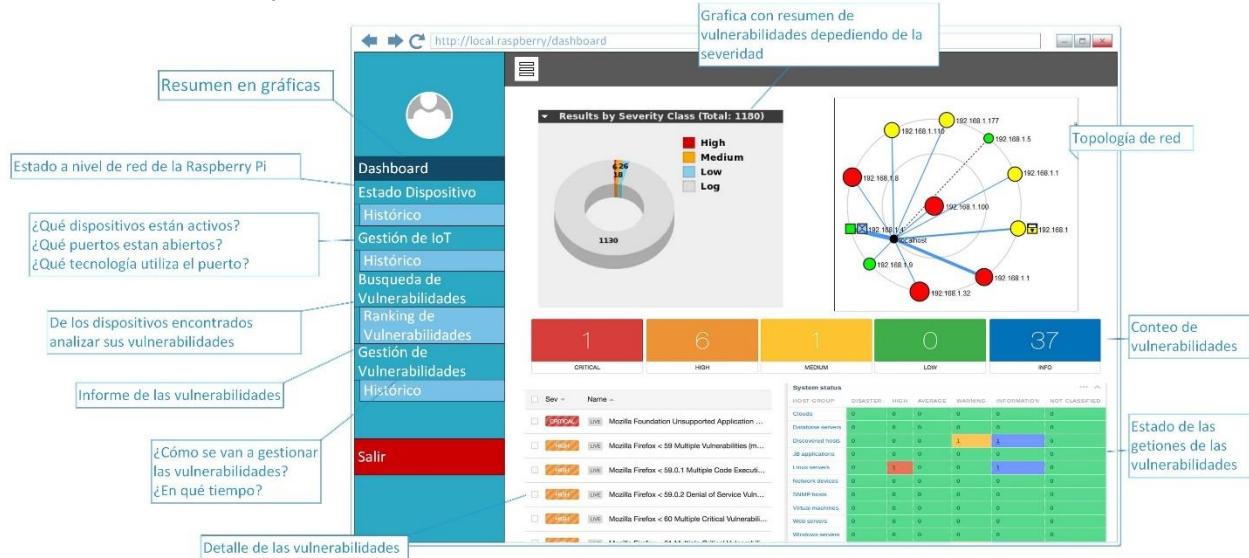
Requisito	Tarea	Quien	Estado	Dia	L	M	X	J	V	L	M	X	J	V	M	X	J	V
				Fecha	19/10/2020	20/10/2020	21/10/2020	22/10/2020	23/10/2020	25/10/2020	27/10/2020	28/10/2020	29/10/2020	30/10/2020	31/10/2020	4/11/2020	5/11/2020	
				Dia Ejecución	98	99	100	101	102	103	104	105	106	107	108	109	110	111
				Horas Pendientes	224	216	208	200	192	184	176	168	160	152	144	136	128	120
Sprint 8																		
Interfaz gráfica del usuario	Gráfica histórico avance de vulnerabilidades	LF	En progreso					5	6									
	Gráfica resumen global de vulnerabilidades	LF	En progreso							6	1	2						
	Gráfica resumen global de dispositivos	LF	En progreso															
	Gráfica resumen global de puertos	LF	En progreso															
	Gráfica resumen gestión de	LF	En progreso															
Modulo dashboard	Gráfica histórico avance de vulnerabilidades	LF	En progreso		3	8	2											
	Gráfica resumen global de vulnerabilidades	LF	En progreso						2	7	4							
	Gráfica resumen global de dispositivos	LF	En progreso							2	2	7						
	Gráfica resumen global de puertos	LF	En progreso															
	Gráfica resumen gestión de	LF	En progreso															
	Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2	LF	En progreso															1

Fuente: Elaboración Propia

Mockups

Se establece de forma tentativa el uso de la siguiente Ilustración 68 como interfaz de usuario para el dashboard.

Ilustración 67. Mockup Dashboard



Fuente: Elaboración Propia

3.2.11.11 Planning Nueve

Para este último sprint se tiene como meta realizar la creación del módulo de estado dispositivo, la integración del API REST de Shodan y el instalador del software; en donde el primer módulo permite visualizar el estado actual del nombre del sistema, dirección IP,

dirección MAC, sistema operativo y fecha/hora del registro en donde se encuentra instalado el software, dando la posibilidad en forma listada de visualizar el estado histórico que permite al usuario del software encontrar la solución de problemas que usualmente se presentan por perdida de conectividad y facilidad de encontrar cambio de red a través del tiempo.

Por otra parte, la integración del motor de búsqueda denominado Shodan, lanzado en el año 2009 por el informático John Matherly, brinda la posibilidad de encontrar dispositivos expuestos en internet tales como router, servidores, cámaras IP, infinidad de objetos IoT, etc. (SHODAN, 2009), dando la posibilidad a un usuario del software conocer que sistemas, servicios o dispositivos tiene expuestos en su organización, bien sea haciendo una búsqueda directa del nombre de la organización, país, sistema operativo, puerto o nombre de sistema, mostrando un reconocimiento general del entorno informático expuesto en internet que se encuentra propenso a ataques informáticos externos.

Es evidente entonces que se convierte en punto de partida frente al análisis del administrador, quien decide si determinado servicio, dispositivo o puerto debe estar expuesto en internet – teniendo presente los vectores y superficie de ataque al que se puede enfrentar cierto sistema con base en a las ciberamenazas que se presentan en la actualidad como hacktivismo, cibercrimen, ciberespionaje, ciberterrorismo o ciberguerra – o debe ser cerrado.

En la siguiente Ilustración 69 se muestra el cronograma del planning nueve.

Ilustración 68. Cronograma Planning

NOMBRE DE TAREA	TIPO DE CARACTERÍSTICA	RESPONSABLE	PUNTUACIÓN HU	INICIO	FINAL	DURACIÓN (DIAS)	ESTADO
Sprint 9				9/11/2020	27/11/2020	14	En progreso
Verificar estado actual dispositivo	Módulo	Luis Felipe N.	3	9/11/2020	10/11/2020	2	En progreso
Histórico de estados de dispositivo		Luis Felipe N.	3	11/11/2020	12/11/2020	2	En progreso
Integración API Shodan	Integración	Luis Felipe N.	3	12/11/2020	12/11/2020	1	En progreso
Buscador shodan		Luis Felipe N.	3	13/11/2020	18/11/2020	3	En progreso
El sistema debe ser capaz de procesar 3 tareas simultaneas	Rendimiento	Luis Felipe N.	3	19/11/2020	19/11/2020	1	En progreso
El sistema debe proporcionar mensajes de error/satisfactorios que sean informativos y orientados a usuario final	Trazabilidad	Luis Felipe N.	3	20/11/2020	24/11/2020	3	En progreso
Mostrar correo de usuario logueado		Luis Felipe N.	1	19/11/2020	20/11/2020	2	En progreso
Desplegar/contrae módulos		Luis Felipe N.	1	25/11/2020	26/11/2020	2	En progreso
Instalador	Despliegue	Luis Felipe N.	1	26/11/2020	27/11/2020	2	En progreso

Fuente: Elaboración Propia

¿Sprint Goal?

Crear el módulo de estado de dispositivo, integración con Shodan, crear instalador y funcionalidades adicionales del software.

Retrospectiva

Finalizado satisfactoriamente el sprint ocho, se observó una responsabilidad individual por parte de cada uno de los integrantes del equipo, teniendo claro las tareas que fueron asignadas en el sprint y a lo largo del desarrollo del proyecto, apoyando las necesidades que se fueran presentando, brindando un ambiente agradable en el cual desenvolverse y sentir un nivel de motivación que ayudara a la culminación de las actividades con calidad y forma organizada, por lo que se mantendrá la dinámica como en anteriores sprints para no afectar procesos y la armonía del equipo.

Desarrollo de Sprint Nueve

En este sprint se prevé la creación del módulo de estado de dispositivo, por lo que será necesario hacer uso de librerías de Python para interacción con el sistema operativo con el que cuente la Raspberry Pi – que de forma nativa para el proyecto será Raspbian basado en arquitectura Linux – para monitorizar aspecto de red. Así mismo se plantea, identificar la estructura del API de Shodan para integrarlo y poder hacer búsqueda de forma personalizada, evitando posibles fallas en el consumo del servicio.

Finalmente, se hará la creación del script en bash para que el proceso de despliegue de la herramienta sea fácil y amigable.

Sprint Backlog

El sprint backlog para este noveno sprint se encuentra sujeto a lo especificado en la siguiente Tabla 60.

Tabla 60. Resumen Sprint Nueve

Nombre de Proyecto	Detección de Vulnerabilidades en Dispositivos IoT
Project Manager	Luis Felipe Naranjo
Ingeniero de Desarrollo	Luis Felipe Naranjo
Número de Sprint	09
Número de Tareas	15
Día Inicio	9/11/2020
Día Final	27/11/2020
Total de Días	14
Progreso General	100%

Fuente: Elaboración Propia

En la siguiente Tabla 61 se evidencia la programación detallada del sprint nueve.

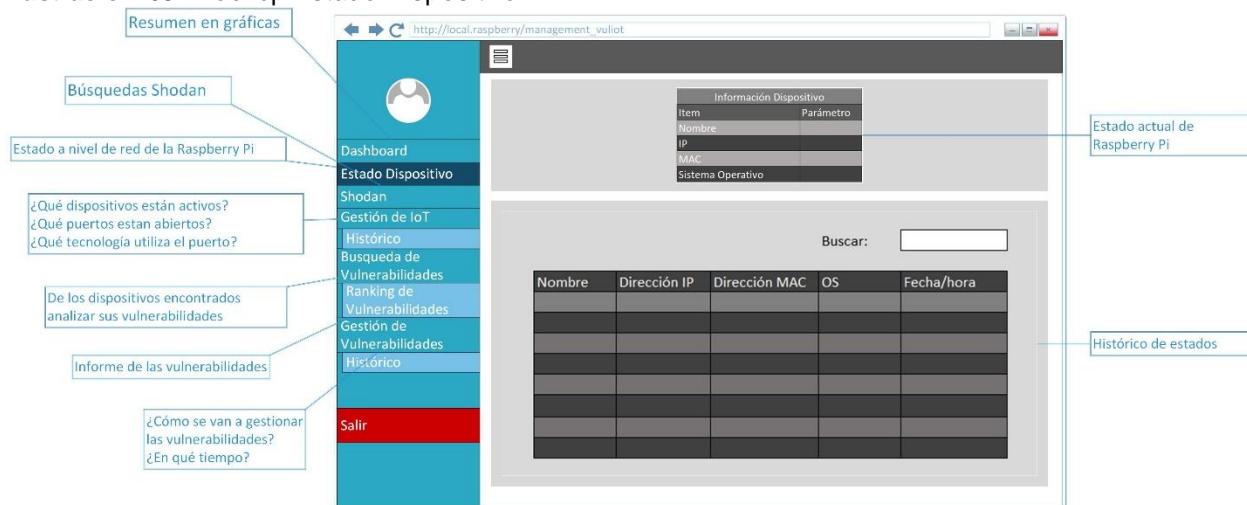
Tabla 61. Programación Sprint Nueve

Requisito	Tarea	Quien	Estado	Dia	L	M	X	J	V	M	X	J	V	L	M	X	J	V
				Fecha	9/11/2020	10/11/2020	11/11/2020	12/11/2020	13/11/2020	17/11/2020	18/11/2020	19/11/2020	20/11/2020	23/11/2020	24/11/2020	25/11/2020	26/11/2020	27/11/2020
				Día Ejecución	112	113	114	115	116	117	118	119	120	121	122	123	124	125
				Horas Pendientes	112	104	96	88	80	72	64	56	48	40	32	24	16	8
Sprint 9																		
Interfaz gráfica del usuario	Verificar estado actual dispositivo	LF	En progreso		6	1												
	Histórico de estados de dispositivo	LF	En progreso				5											
	Buscador shodan	LF	En progreso							8	3	2						
	Mostrar correo de usuario logueado	LF	En progreso									3	2					
	Desplegar/contraer módulos	LF	En progreso													6	4	
Modulo estado de dispositivo	Verificar estado actual dispositivo	LF	En progreso		2	7												
	Histórico de estados de dispositivo	LF	En progreso				3	5										
Modulo Shodan	Integración API Shodan	LF	En progreso					3										
	Buscador shodan	LF	En progreso							5	6							
Usabilidad y rendimiento	El sistema debe ser capaz de procesar 3 tareas simultaneas	LF	En progreso									4						
	El sistema debe proporcionar mensajes de error/satisfactorios que sean informativos y orientados a usuario final	LF	En progreso										1	8	2			
	Mostrar correo de usuario logueado	LF	En progreso								1	5				2	2	
	Desplegar/contraer módulos	LF	En progreso													2	8	
	Instalador	LF	En progreso															

Fuente: Elaboración Propia

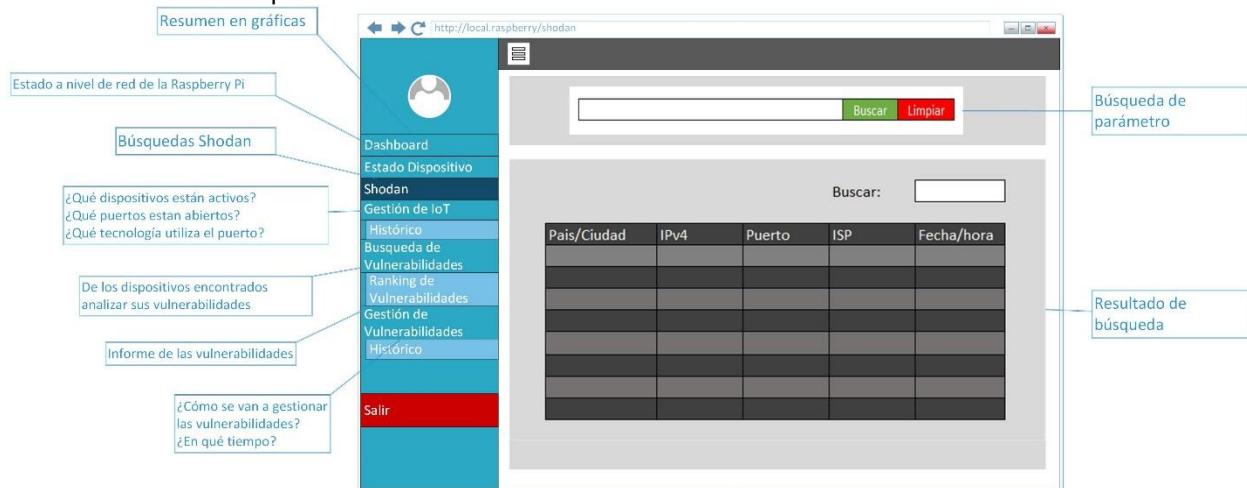
Mockups

Se establece de forma tentativa el uso de la siguiente Ilustración 70 como interfaz de usuario para mostrar el estado del dispositivo.

Ilustración 69. Mockup Estado Dispositivo

Fuente: Elaboración Propia

Se observa en la Ilustración 71 el mockup correspondiente a las búsquedas integrando el API de Shodan.

Ilustración 70. Mockup Shodan

Fuente: Elaboración Propia

3.2.12 Características y Funcionalidades del Software

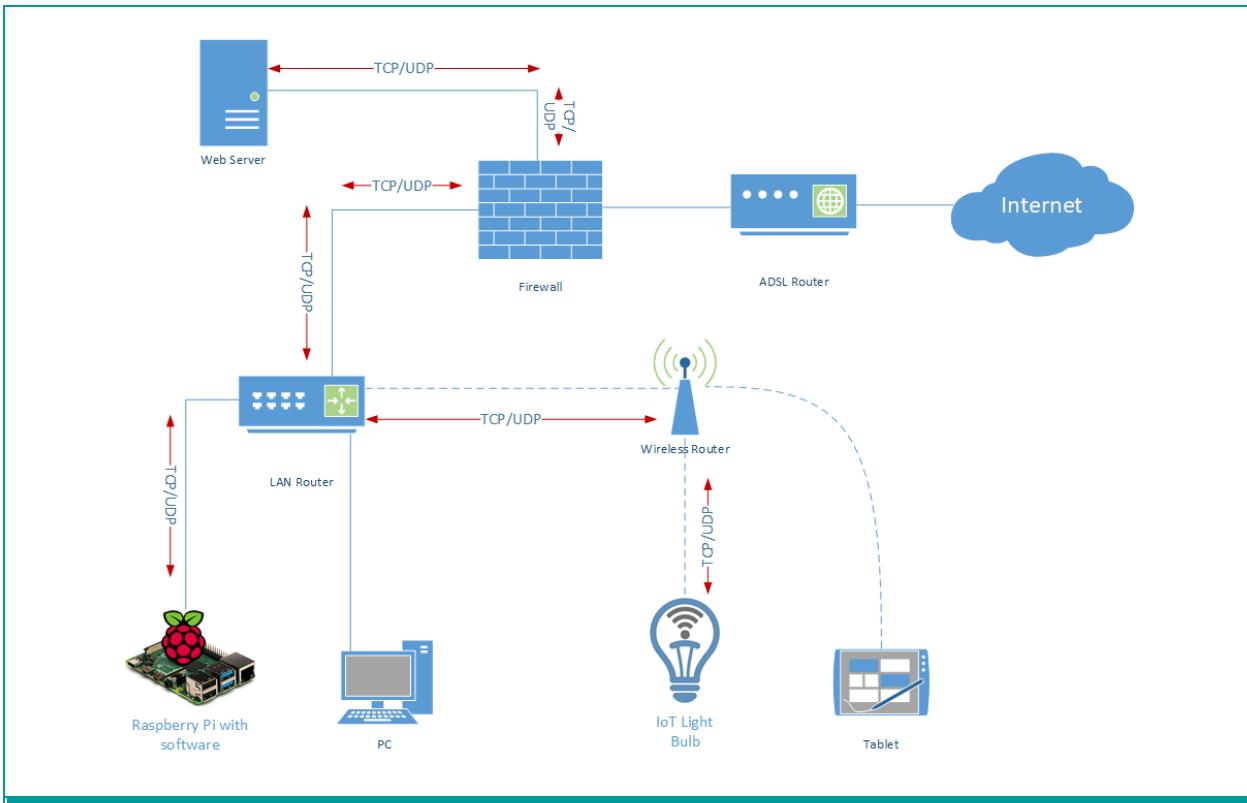
La integración del método de escaneo y método de detección de vulnerabilidades se convierten en el complemento adecuado para convertirse en una herramienta de software conveniente para el descubrimiento de infraestructuras tecnológicas, ayudando al proceso de eliminación y mitigación de vulnerabilidades que brinda al usuario un nivel de evolución de criterios para la toma de decisiones a la hora de resguardar activos tecnológicos frente ciberamenazas, en donde el software cumple características las cuales brindan:

- **Exactitud:** Este es uno de los criterios más importantes frente al desarrollo del software, teniendo un grado de concordancia entre el resultado de una detección de vulnerabilidad y el nivel de clasificación/valoración cuantitativa realizada de forma interna en el software, se obtiene un resultado global del estado de uno o un grupo de dispositivos. Brindando la seguridad que se tendrán resultados lo más completos posibles acorde a lo descubierto en un dispositivo, disminuyendo falsos positivos o información errónea. Tomando como base lo descrito en la ISO 572-1:1994 (ISO, 1994) denominada Exactitud (veracidad y precisión) de los métodos y resultados de medición, teniendo presente **Exactitud = Veracidad + Precisión**.
- **Usabilidad:** A pesar que se comprende en un marco de software no altamente utilizado por usuario no técnicos en aspectos tecnológicos, su nivel de comprensión, secuencialidad en cada una las tareas y facilidad de lectura de resultados en compañía de gráficas, se convierte en una herramienta apta para la detección de amenazas en dispositivos IoT, asegurando que el usuario del software pueda tomar decisiones pertinentes con base en el entorno tecnológico en el que se encuentre. Brindando facilidad y métodos intuitivos de acoplamiento, agilizando el desarrollo de actividades e impidiendo la pérdida de tiempo a causa de metodologías poco ortodoxas, permitiendo con unos clics operar el software sin largas instrucciones y documentación externa.

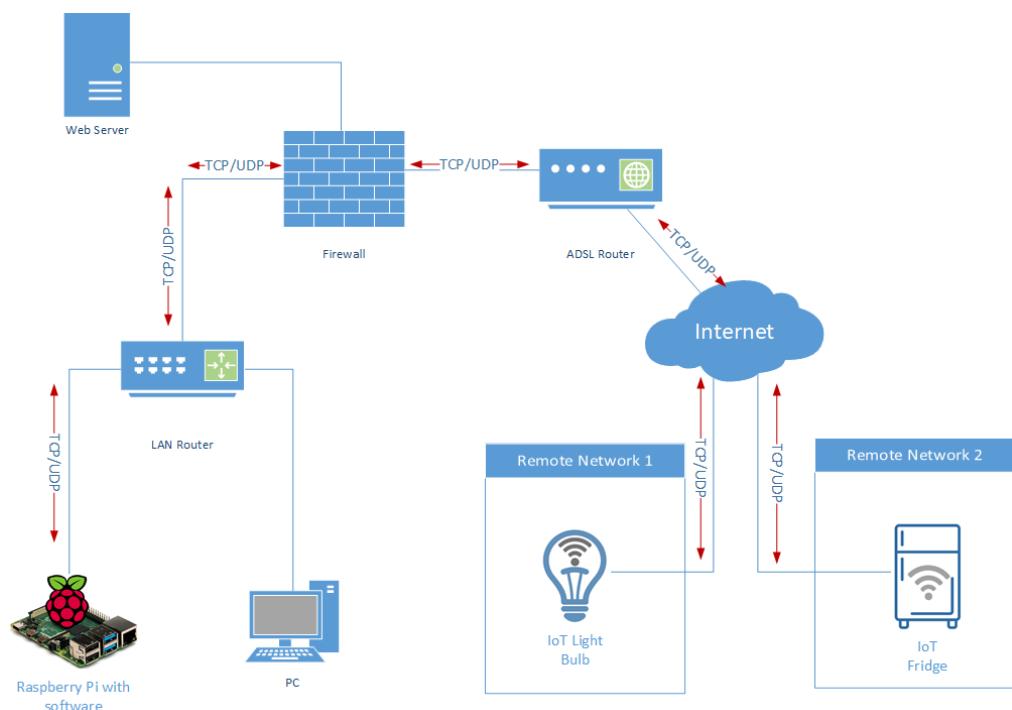
- **Seguridad:** El uso de usuarios con credenciales de acceso privadas y el proceso de almacenamiento del software, aseguran que todos los escaneos y detecciones realizadas por un usuario sean accesibles por el usuario correspondiente, permitiendo mantener en todo momento la confidencialidad, integridad y la autenticidad de la información recolectada.
- **Integración y administración:** La facilidad de integración en sistemas operativos basados en Linux permite un rendimiento óptimo en el uso de las herramientas internas del sistema, garantizando no solo resultados rápidos entre tareas sino un consumo de recursos bajo, haciendo que la estación de trabajo donde se ejecuta el software no presente perdida de recursos en memoria RAM, CPU, disco duro, disco de estado sólido o consumo excesivo de corriente. Así mismo, su proceso de administración es sencilla, permitiendo ponerlo en marcha solo con la ejecución del script de instalación.
- **Uso en Entorno Internos/Externos:** Gracias a la interacción permitida por el software, este puede realizar el intercambio de información con entornos externos o conocidos como sistemas expuestos a internet, lo que ayuda a un usuario que esté haciendo una auditoría informática – dentro del marco legal – realizar descubrimiento de información a nivel de red y vulnerabilidades, que puede ser fundamental para entender sistemas propensos a amenazas. Este proceso de interacción tanto interna como externa se muestra en la siguiente Tabla 62.

Tabla 62. Usabilidad Entornos Privados o Expuestos en Internet

Uso en Entornos Privados



Uso en Entornos Expuestos a Internet



Fuente: Elaboración Propia

4. PRUEBAS Y RESULTADOS

Contando con un producto de software integrado con los requerimientos óptimos que satisfacen las necesidades finales de uso, es necesario realizar pruebas que forma general muestren un correcto funcionamiento, un proceso de tratamiento adecuada de errores y resultados acordes a lo esperado. Integrando en paralelo aspectos de funcionamiento interno con una interfaz gráfica que sea agradable al usuario final, probadas de forma unitaria en el desarrollo de cada sprint, en este apartado del proyecto se muestran las diferentes pruebas con sus resultados, demostrando la incorporación completa de módulos y el nivel de alcance en el que puede ser usado el software desarrollado en ámbitos individuales y organizacionales.

4.1 ALCANCE DE PRUEBAS

De forma general el software desarrollado se compone de módulos integrados, por lo que a continuación se presenta el alcance que se pretende como meta para el cumplimiento de criterios de aceptación.

4.1.1 Resumen de Pruebas

En la siguiente Tabla 63 se establece el resumen general de pruebas.

Tabla 63. Resumen de Pruebas

Módulos del sistema a ser probados:	Todos
Objetivos de las pruebas:	<p>Validar:</p> <ul style="list-style-type: none"> • Comportamiento de la herramienta con el ingreso, modificación y eliminación de datos en el sistema • Secuencia lógica y ordenada de las funcionalidades • Manejo general de errores • Resultados claros, concisos y acordes a la realidad del sistema objetivo
Orden de ejecución:	<p>Los módulos son secuenciales, por lo que el modelo de ejecución se comporta:</p> <ul style="list-style-type: none"> • Gestión de IoT • Scan Vulnerabilidades • Gestión de Vulnerabilidades
Responsable de las pruebas:	<p>Limitado por el número de personal dedicado al proyecto, se destina que el mismo desarrollador del software tome el rol de tester, con una perspectiva propia al cargo y análisis crítico que ayuden al desarrollo correcto de las pruebas:</p> <p>Responsable: Luis Felipe N.</p>
Fecha Ejecución:	1 de Abril/2021 – 2 de Abril/2021

Fuente: Elaboración Propia

4.2 ENTORNO Y CONFIGURACIÓN DE LAS PRUEBAS

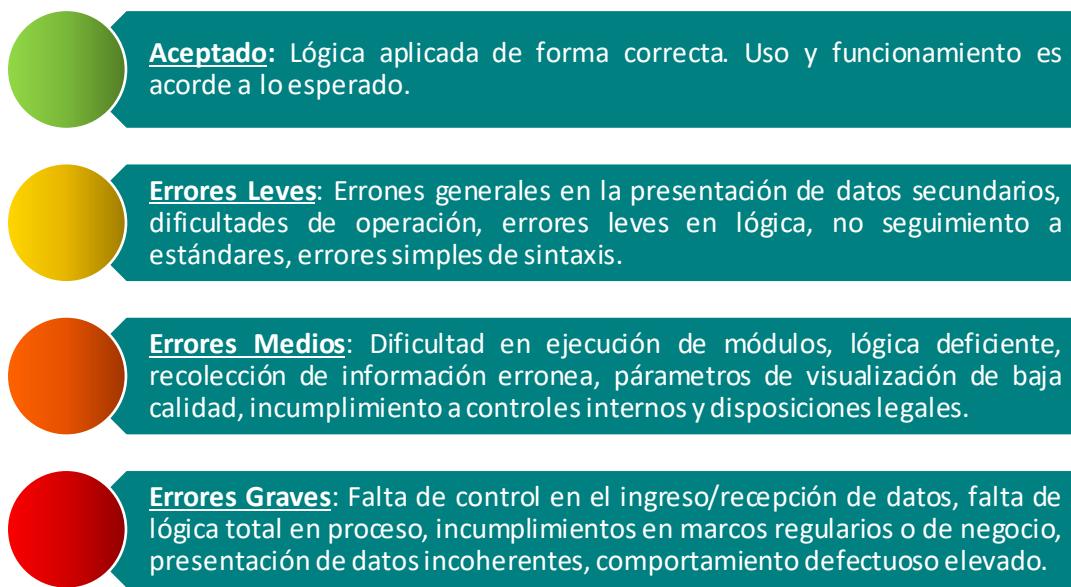
Para el proceso de ejecución de las pruebas del software se requiere de la disponibilidad de los siguientes elementos:

- Equipo sobre el cual se hará la instalación del Software:
 - Raspberry Pi OS, GNU/Linux basado en Debian, versión Buster
 - Raspberry PI 4 Model B, procesador 64-bit quad-core Cortex-A72, 8 GB LPDDR4 RAM, memoria micro SD 50 GB, 802.11b/g/n/ac inalámbrico, puerto 802.3 Gigabit Ethernet, puerto micro HDMI, 5V/3A USB-C
- Equipo objetivo:
 - Software base desarrollado de forma individual
 - ESP32, microprocesador de 32-bit Xtensa LX6, memoria 520 KiB SRAM, 802.11b/g/n inalámbrico, bluetooth v4.2 BR/EDR y BLE, LED PWM (hasta 16 canales)

4.3 CRITERIOS DE APROBACIÓN

Para la correcta asignación de conformidad con el resultado en las pruebas, se hará uso del siguiente modelo de clasificación observado en la Ilustración 72.

Ilustración 71. Clasificación Control de Calidad de Software



Fuente: Tomado de <https://bit.ly/3ePHAz>

4.4 ESTRATEGIA Y PLAN DE EJECUCIÓN DE PRUEBAS

Para llevar una ejecución ordenada del proceso de pruebas sobre el software desarrollado, se ha planteado enmarcar el siguiente plan de ejecución que se puede observar en la Tabla 64.

Tabla 64. Estrategia y Plan de Pruebas

Fase	Proceso de Prueba	Enfoque	Tipo de Prueba	Sub-tipo de Prueba	Módulos de Software	Responsable
I	Dinámica	Caja Negra	Funcional	Entorno Privado	Gestión de IoT Scan Vulnerabilidades	Luis Felipe N.
				Entorno Público		
II	Dinámica	Caja Negra	Funcional	Unitaria	Todos	Luis Felipe N.
				Humo		
				Integración		
				Funcional		
			No Funcional	Interfaz		

Fuente: Elaboración Propia

4.5 DESARROLLO DE PRUEBAS Y RESULTADOS

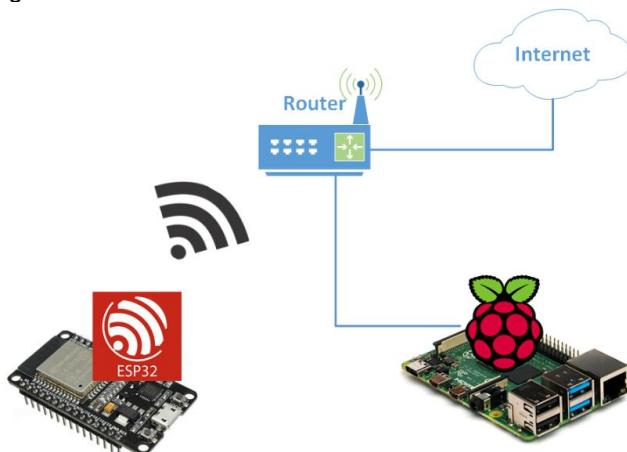
En este apartado se realiza la ejecución correspondiente de las pruebas al software de forma general en entornos controlados, identificando que el comportamiento deseado sea el adecuado siguiendo la lógica aplicada en su momento de codificación.

4.5.1 Desarrollo Fase I de Pruebas

4.5.1.1 Pruebas en Entorno Privado

Titulo	LAN – Entorno Privado		
Prueba N°	01	Versión	v1
Fecha	1 de Abril/2021	Duración	30 Min
Estado Aprobación	Aceptado		
Módulo del Sistema	- Gestión de IoT - Scan Vulnerabilidades	Dispositivos	- Raspberry PI 4 Model B - ESP32
Objetivo	Detectar vulnerabilidades existentes en tarjeta IoT ESP 32 en un ambiente privado controlado		
Datos de Entrada	Dirección IPv4		
Topología			

Ilustración 72. Topología Pruebas en Entorno Privado

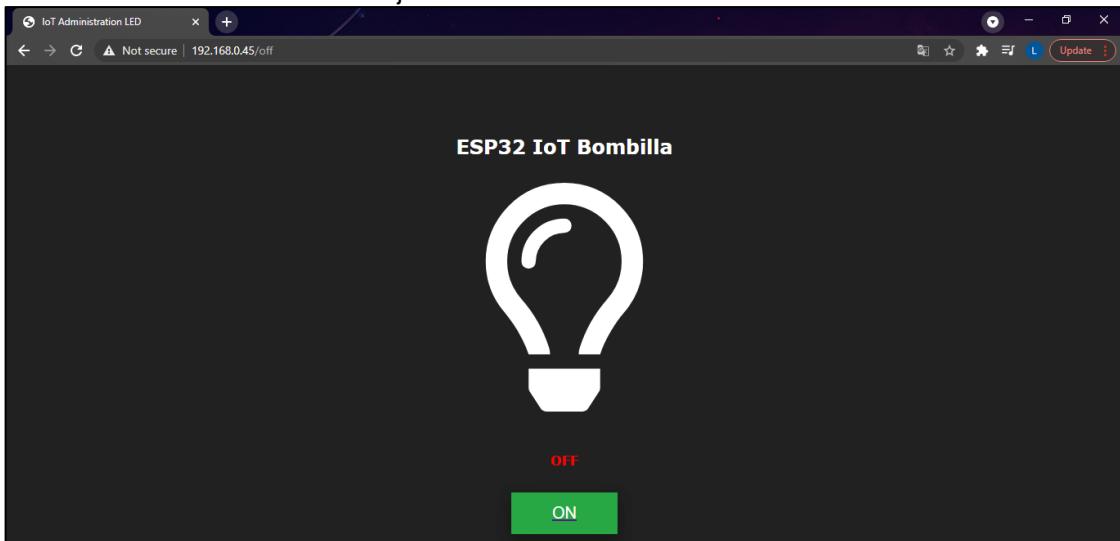


Fuente: Elaboración Propia

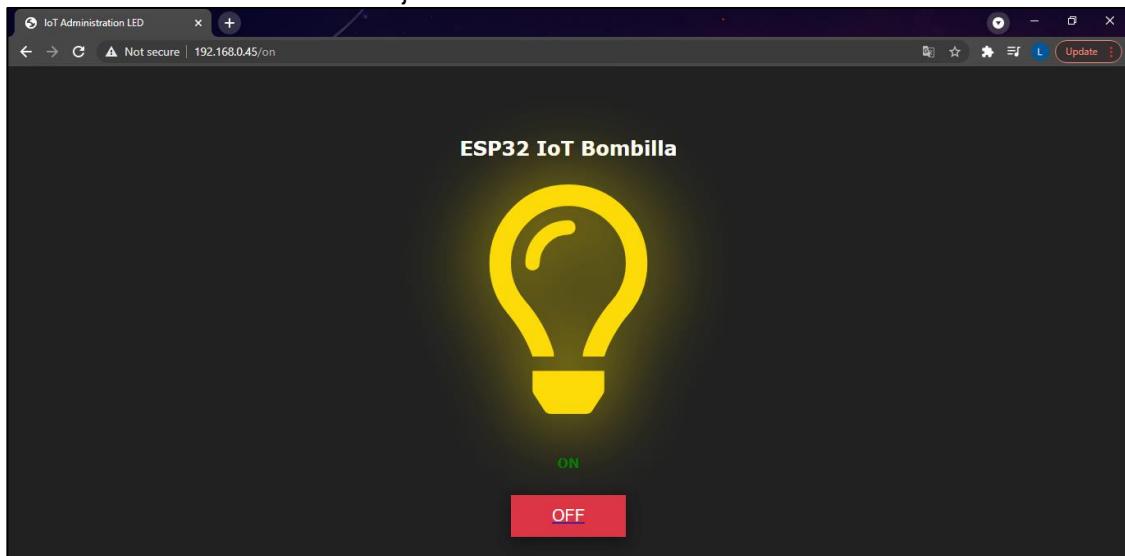
Pasos de Ejecución

- 1) Programada la tarjeta IoT ESP32 con un software para control de iluminación de una bombilla LED con integración básica de control – como se muestra en la Ilustración 74 e Ilustración 75 - permite emular en ambiente controlado un ciber-ataque a este dispositivo mediante el esquema de software previamente configurado en la Raspberry Pi, realizando un ataque automatizado, sin intervención de factores externos y de forma aislada, garantizando así la toma de resultados acertados.

Ilustración 73. Simulación de Objeto IoT I

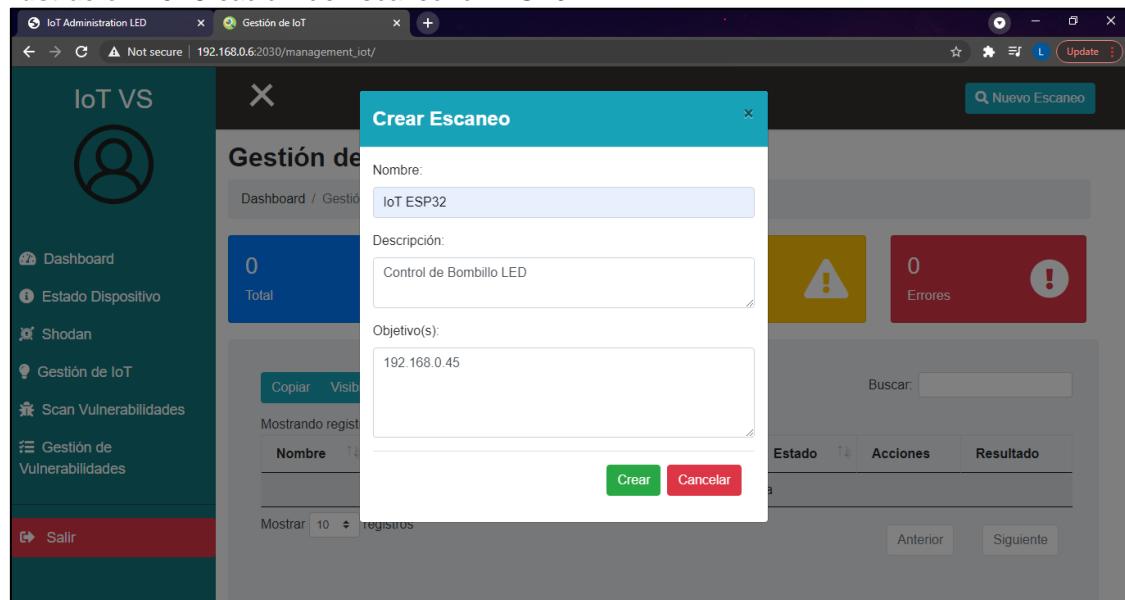


Fuente: Elaboración Propia

Ilustración 74. Simulación de Objeto IoT II

Fuente: Elaboración Propia

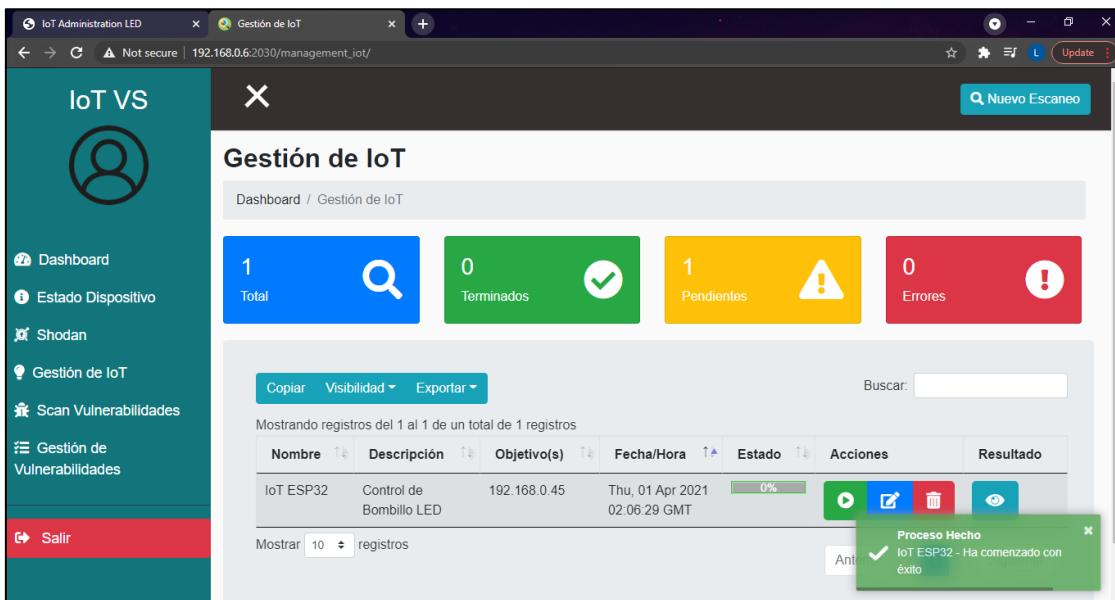
- 2) En el esquema de software implementado en Raspberry Pi, en primera instancia se procede a realizar el descubrimiento de la tarjeta IoT ESP32, para ello, en la opción denominada “Gestión de IoT” se procede a crear un nuevo escaneo con los parámetros correspondientes propios al dispositivo IoT como se muestra en la Ilustración 76. Teniendo en conocimiento la dirección IP objetivo, se procede a determinar de forma precisa en la creación del escaneo para evitar un escaneo completo a la red que conlleve demoras.

Ilustración 75. Creación de Escaneo IoT ESP32

Fuente: Elaboración Propia

- 3) Se inicia proceso de descubrimiento de dispositivo IoT de acuerdo a los parámetros establecidos – Ilustración 77.

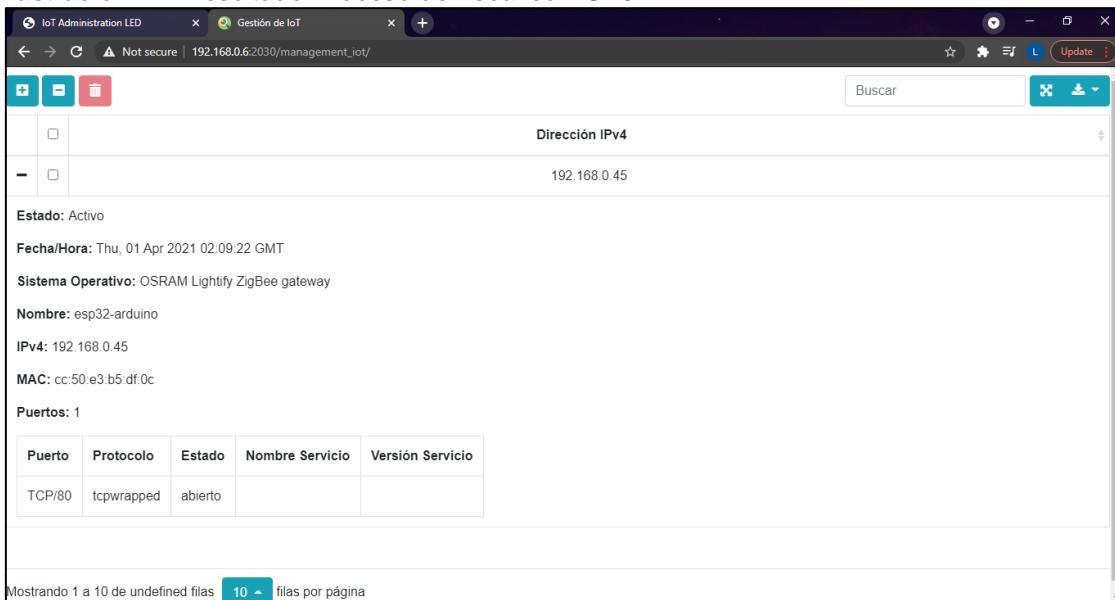
Ilustración 76. Inicio de Proceso de Descubrimiento



Fuente: Elaboración Propia

- 4) Teniendo como resultado un 100% en el proceso de descubrimiento del dispositivo IoT ESP32 que incluye dirección MAC, sistema operativo, nombre, estado, número de puertos y servicios – como se muestra en la Ilustración 78 - se puede detallar información del elemento encontrado un resultado.

Ilustración 77. Resultado Proceso de Escaneo ESP32

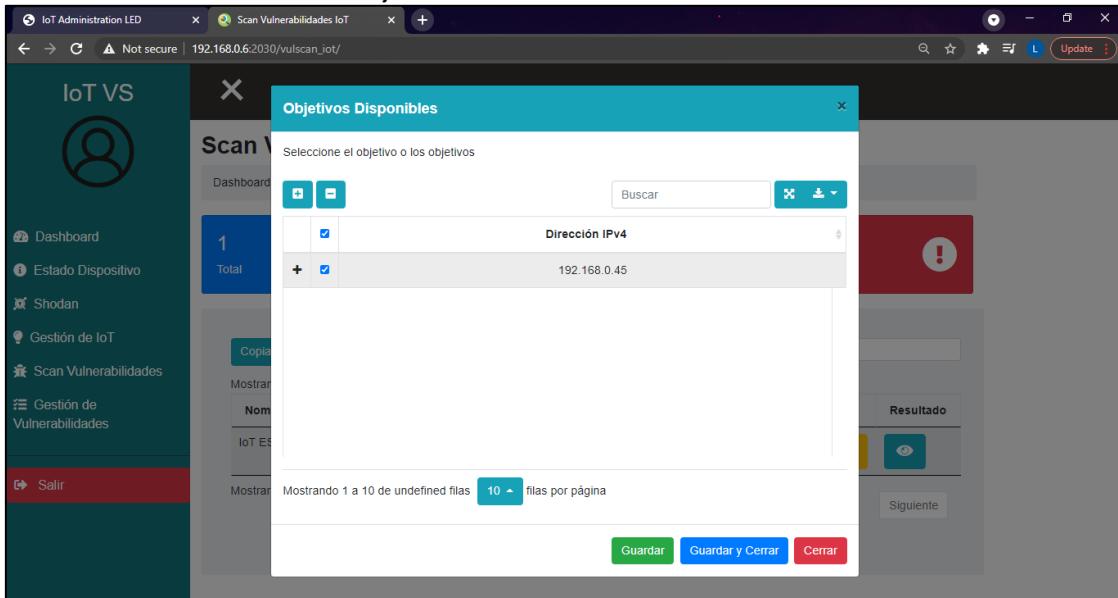


Fuente: Elaboración Propria

- 5) Teniendo presente parámetros propios de la tarjeta IoT ESP32 descubiertos por el esquema de software implementado en la Raspberry Pi, ahora en el apartado denominado “Scan Vulnerabilidades” se procede en primera instancia a seleccionar

el objetivo como se muestra en la Ilustración 79, que para esta prueba solo se encuentra disponible la tarjeta ESP32.

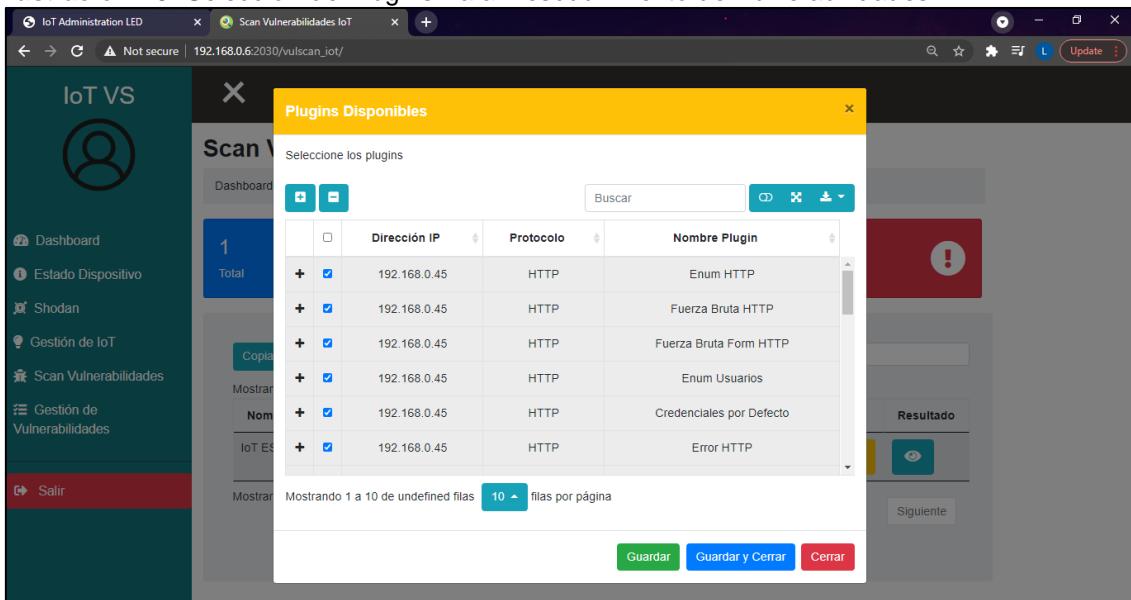
Ilustración 78. Selección de Objetivo Para Detección de Vulnerabilidades



Fuente: Elaboración Propia

- 6) Descubierto por módulo “Gestión de IoT” el puerto TCP/80 correspondiente a HTTP en la tarjeta IoT ESP32 el software de forma automática muestra plugins correspondientes a este servicio, que permitirán hacer el descubrimiento de vulnerabilidades. Para efectos de prueba se seleccionarán todos los plugins disponibles como se evidencia en la Ilustración 80.

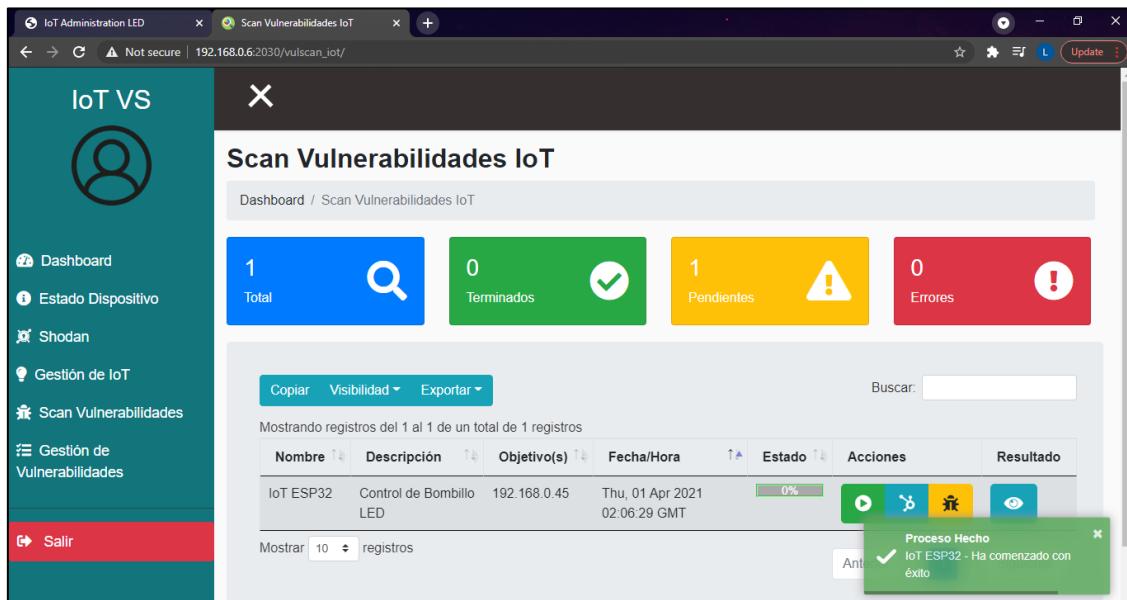
Ilustración 79. Selección de Plugins Para Descubrimiento de Vulnerabilidades



Fuente: Elaboración Propia

- 7) Se inicia proceso de detección de vulnerabilidades en dispositivo IoT de acuerdo a los parámetros establecidos – Ilustración 81.

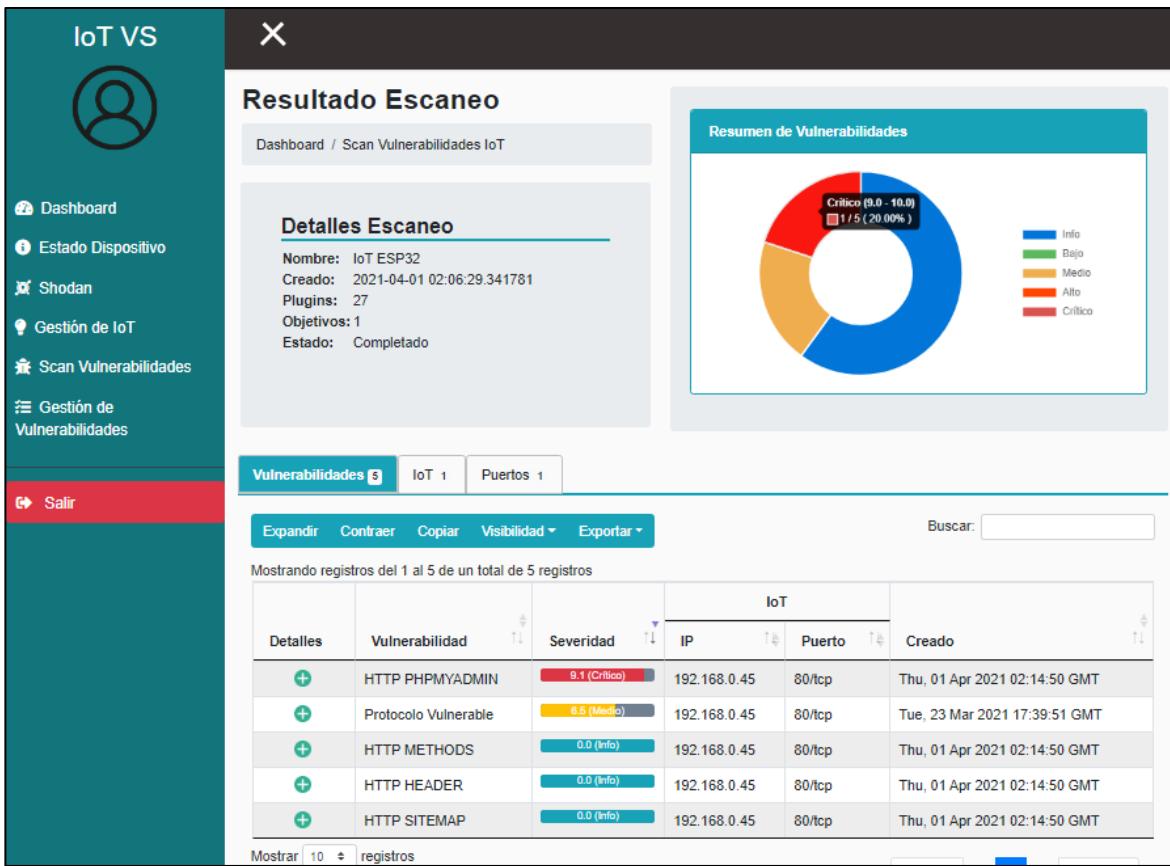
Ilustración 80. Inicio Proceso de Detección de Vulnerabilidades



Fuente: Elaboración Propia

Resultados

En el resultado obtenido de la prueba automatizada de ciber-ataque sobre la tarjeta IoT ESP32 se encuentra el uso de 27 Plugins, en donde se detectaron 5 hallazgos que pueden ayudar a identificar problemas a nivel seguridad de software sobre este dispositivo – como se muestra en la Ilustración 82.

Ilustración 81. Resultado Escaneo de Vulnerabilidades

Fuente: Elaboración Propia

El conglomerado completo cuenta con hallazgos de nivel informativos, medios y críticos, cada uno identificado de forma clara, con su respectiva descripción, resultado e información que ayuda al usuario final a la toma de decisiones y/o proceso adecuado de remediación sobre la vulnerabilidad como se muestra en la Ilustración 83.

Ilustración 82. Resumen Hallazgos Encontrados

Detalles	Vulnerabilidad	Severidad	IP	Puerto	Creado
+	HTTP PHPMYADMIN	9.1 (Critical)	192.168.0.45	80/tcp	Thu, 01 Apr 2021 02:14:50 GMT
x	Protocolo Vulnerable	6.5 (Medio)	192.168.0.45	80/tcp	Tue, 23 Mar 2021 17:39:51 GMT

Mostrando registros del 1 al 5 de un total de 5 registros

Resumen:

Servicios de Red Inseguros

Resultado de Vulnerabilidad:

Protocolo tcp/80 inseguro

Remediación de Vulnerabilidad:

Se recomienda asegurar los servicios de red de acuerdo a: 1. Asegurar que solo los puertos necesarios estén expuestos y disponibles 2. Hacer uso de protocolos seguros como SSH 3. Garantizar que los servicios no sean vulnerables al desbordamiento del búfer y a los ataques de fuzzing. 4. Garantizar que los servicios no sean vulnerables a los ataques DoS. 5. Asegurar que los puertos o servicios de red no estén expuestos a Internet a través de UPnP, por ejemplo.

+	HTTP METHODS	0.0 (Info)	192.168.0.45	80/tcp	Thu, 01 Apr 2021 02:14:50 GMT
+	HTTP HEADER	0.0 (Info)	192.168.0.45	80/tcp	Thu, 01 Apr 2021 02:14:50 GMT
+	HTTP SITEMAP	0.0 (Info)	192.168.0.45	80/tcp	Thu, 01 Apr 2021 02:14:50 GMT

Mostrar 10 registros

Fuente: Elaboración Propia

Hallazgos Informativos

- 1) **HTTP METHODS:** Métodos HTTP identificados en respuesta a peticiones especialmente creadas. Siendo los mostrados en la Ilustración 84 los más comunes y de bajo riesgo.

Ilustración 83. Resultado HTTP METHODS ESP32

x	HTTP METHODS	0.0 (Info)	192.168.0.45	80/tcp	Thu, 01 Apr 2021 02:14:50 GMT
-------------------	--------------	------------	--------------	--------	-------------------------------

Resumen:

Informativo

Resultado de Vulnerabilidad:

Supported Methods: GET HEAD POST OPTIONS

Remediación de Vulnerabilidad:

Ninguna acción es requerida

Fuente: Elaboración Propia

- 2) **HTTP HEADER:** Permite identificar el navegador del cliente por medio del requests y responses de una petición HTTP, como se observa en la Ilustración 85 la respuesta es básica dado que la petición es simple y no hace uso de un User-Agent con el que cuentan los navegadores de uso normal.

Ilustración 84. Resultado HTTP HEADER ESP32

	HTTP HEADER	0.0 (Info)	192.168.0.45	80/tcp	Thu, 01 Apr 2021 02:14:50 GMT
Resumen:					
Informativo					
Resultado de Vulnerabilidad:					
Content-type:text/html Connection: close (Request type: GET)					
Remediación de Vulnerabilidad:					
Ninguna acción es requerida					

Fuente: Elaboración Propia

3) HTTP SITEMAP: Identifica rutas relevantes en el servidor web – Ilustración 86.

Ilustración 85. Resultado HTTP SITEMAP ESP32

	HTTP SITEMAP	0.0 (Info)	192.168.0.45	80/tcp	Thu, 01 Apr 2021 02:14:50 GMT
Resumen:					
Informativo					
Resultado de Vulnerabilidad:					
Directory structure: / Other: 1 Longest directory structure: Depth: 0 Dir: / Total files found (by extension): Other: 1					
Remediación de Vulnerabilidad:					
Ninguna acción es requerida					

Fuente: Elaboración Propia

Hallazgos Medios

- 1) **Protocolo Vulnerable:** El modelo de software diseñado e implementado en la tarjeta IoT ESP32 no contempla el uso de formularios y/o ingreso a sesiones de autenticación, sin embargo, dada la inseguridad, fácil intercepción y manipulación de comunicaciones que presenta el puerto TCP/80 HTTP, el software de forma automática detecta este puerto y lo categoriza en hallazgos de nivel medio como se evidencia en la Ilustración 87.

Ilustración 86. Resultado Protocolo Vulnerable

	Protocolo Vulnerable	6.5 (Medio)	192.168.0.45	80/tcp	Tue, 23 Mar 2021 17:39:51 GMT
Resumen:					
Servicios de Red Inseguros					
Resultado de Vulnerabilidad:					
Protocolo tcp/80 inseguro					
Remediación de Vulnerabilidad:					
Se recomienda asegurar los servicios de red de acuerdo a: 1. Asegurar que solo los puertos necesarios estén expuestos y disponibles 2. Hacer uso de protocolos seguros como SSH 3. Garantizar que los servicios no sean vulnerables al desbordamiento del búfer y a los ataques de fuzzing. 4. Garantizar que los servicios no sean vulnerables a los ataques DoS. 5. Asegurar que los puertos o servicios de red no estén expuestos a Internet a través de UPnP, por ejemplo.					

Fuente: Elaboración Propia

Hallazgos Críticos

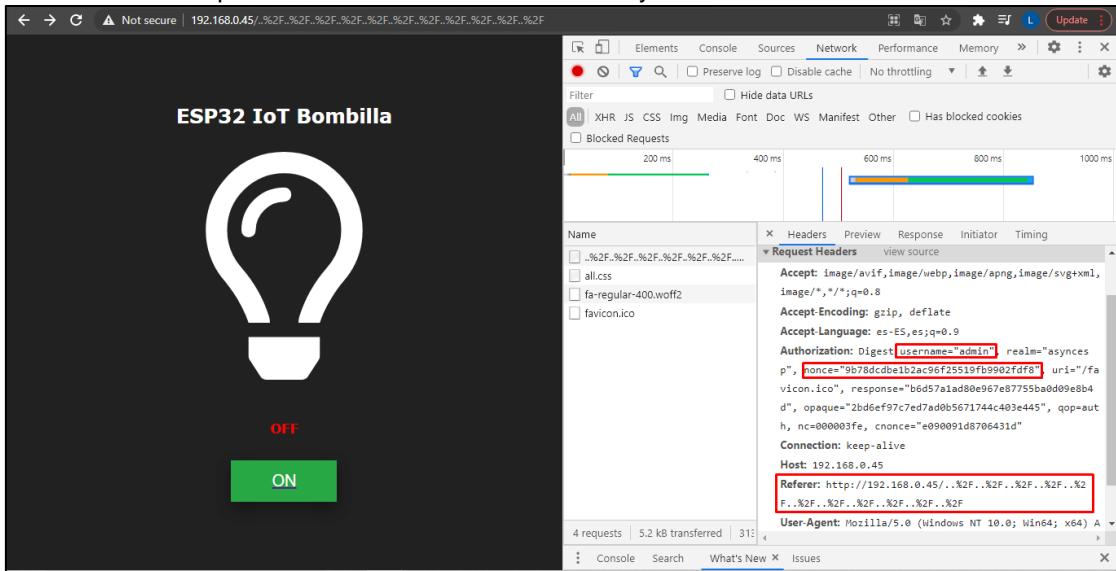
- 1) **HTTP PHPMYADMIN:** A pesar que la tarjeta IoT ESP32 no cuenta con el programa PHP MYADMIN, el software que realiza la detección de vulnerabilidades encuentra que de igual forma la página es vulnerable a Directory Traversal (ver Ilustración 88), haciendo uso del payload ../../../../../../etc/passwd, por lo que no se realiza una validación de información ingresa del lado del cliente, dando paso al acceso de información sensible, incluyendo así código y datos de la aplicación, credenciales para sistemas back-end y archivos confidenciales del sistema operativo.

Ilustración 87. Resultado HTTP PHPMYADMIN ESP32

The screenshot shows a network analysis or penetration testing interface. At the top, there's a header with tabs: 'HTTP PHPMYADMIN' (selected), '9.1 (Crítico)', '192.168.0.45', '80/tcp', and the date 'Thu, 01 Apr 2021 02:14:50 GMT'. Below the header, a section titled 'Resumen:' states 'Protección de Privacidad Insuficiente'. Under 'Resultado de Vulnerabilidad:', it says 'VULNERABLE: phpMyAdmin grab_globals.lib.php subform Parameter Traversal Local File Inclusion State: VULNERABLE (Exploitable) IDs: CVE-CVE-2005-3299 PHP file inclusion vulnerability in grab_globals.lib.php in phpMyAdmin 2.6.4 and 2.6.4-p1 allows remote attackers to include local files via the \$_redirect parameter, possibly involving the subform array. Disclosure date: 2005-10-01 Extra information: ../../../../../../etc/passwd :'. Below this, there's a large lightbulb icon with the word 'ESP32 IoT Bombilla' above it. Underneath the lightbulb is a red button labeled 'ON' and a white button labeled 'OFF'. At the bottom, there's a 'References:' section with links to CVE-2005-3299 and exploit-db.com/exploits/1244. A 'Remediaciación de Vulnerabilidad:' section at the bottom right suggests checking for updates from the provider.

Fuente: Elaboración Propia

El payload se han encodeado para su interpretación y explotación de vulnerabilidad, dando como resultado ..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F el cual al momento de hacer uso contra la pagina se obtiene el resultado que se muestra en Ilustración 89, mostrando información sensible de usuarios, hashes, tipo de encoding, etc.

Ilustración 88. Explotación de Vulnerabilidad Directory Traversal ESP32

Fuente: Elaboración Propia

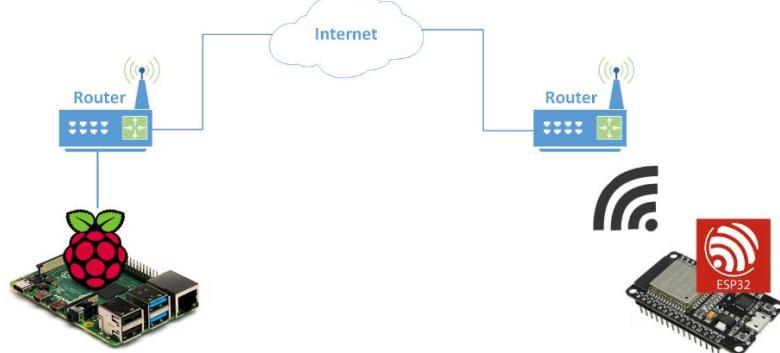
Observaciones

La velocidad en el descubrimiento y detección de vulnerabilidades en el objetivo es directamente proporcional a la tarjeta Raspberry Pi y latencia de comunicación que se encuentre entre el origen y destino de la comunicación.

4.5.1.2 Entorno Público

Título	WAN – Entorno Público		
Prueba N°	02	Versión	V1
Fecha	2 de Abril/2021	Duración	30 Min
Modulo del Sistema	Aceptado		
Módulo del Sistema	- Gestión de IoT - Scan Vulnerabilidades	Dispositivos	- Raspberry Pi 4 Model B - ESP32
Dispositivo	Raspberry PI 4 Model B (8 GB RAM)		
Objetivo	Detectar vulnerabilidades existentes en tarjeta IoT ESP 32 en un ambiente público controlado		
Datos de Entrada			
Dirección IPv4			
Topología			

Ilustración 89. Topología Pruebas en Entorno Público

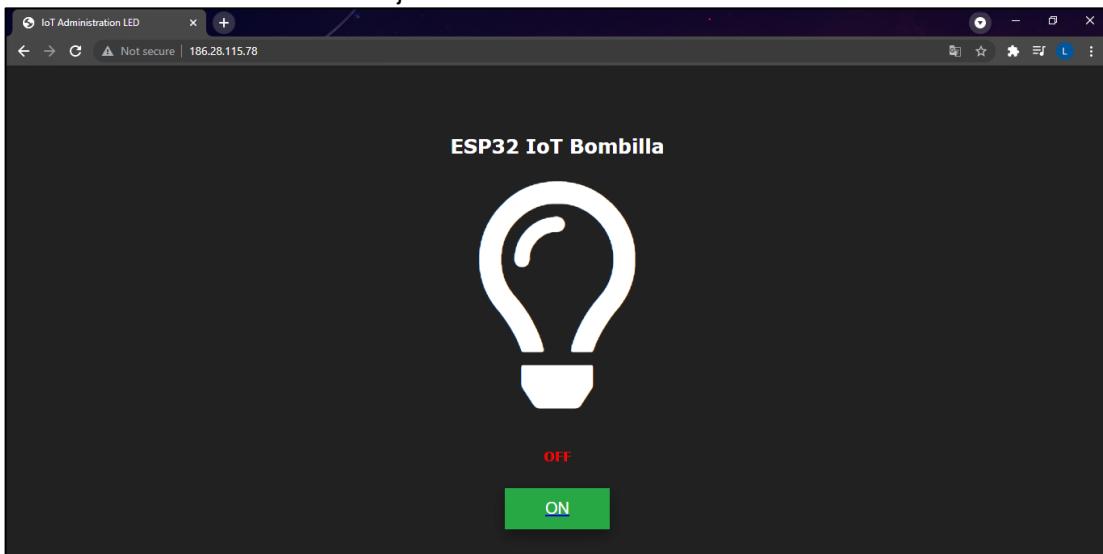


Fuente: Elaboración Propia

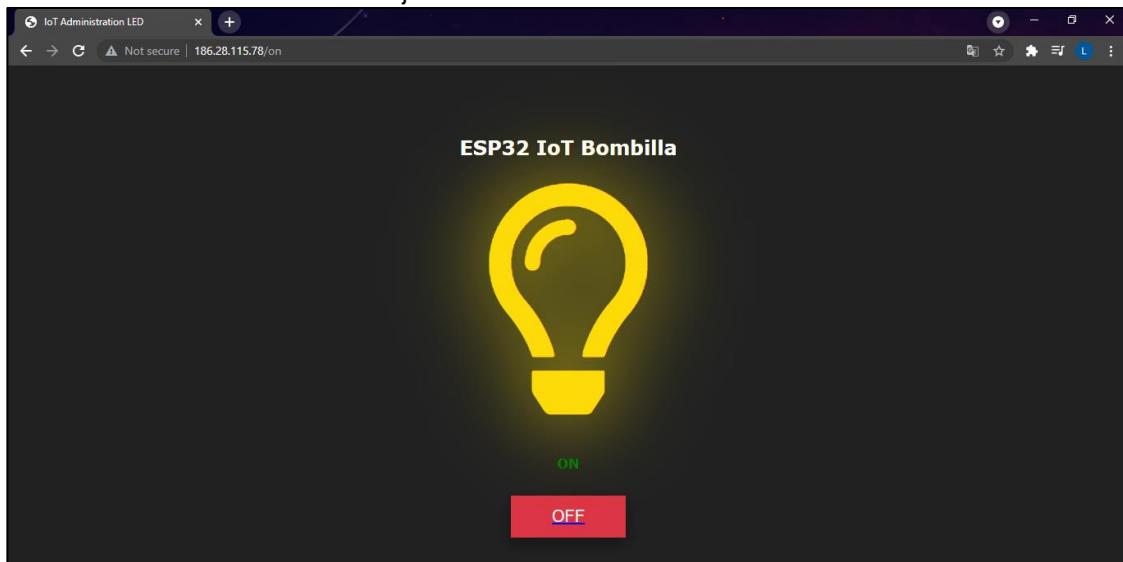
Pasos de Ejecución

- 1) Programada la tarjeta IoT ESP32 con un software para control de iluminación de una bombilla LED con integración básica de control – como se muestra en la Ilustración 91 e Ilustración 92 -, permite emular en ambiente controlado un ciber-ataque a este dispositivo mediante el esquema de software previamente configurado en la Raspberry Pi, realizando un ataque automatizado en entorno público, sin intervención de factores externos y de forma aislada, garantizando así la toma de resultados acertados.

Ilustración 90. Simulación de Objeto IoT I

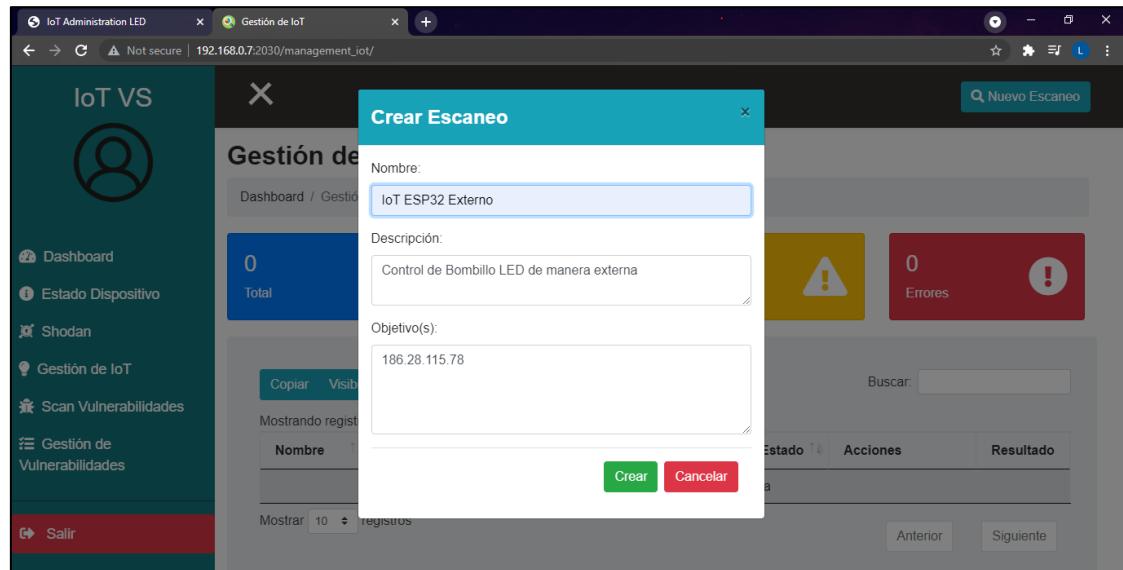


Fuente: Elaboración Propia

Ilustración 91. Simulación de Objeto IoT II

Fuente: Elaboración Propia

- 2) En el esquema de software implementado en Raspberry Pi, en primera instancia se procede a realizar el descubrimiento de la tarjeta IoT ESP32, para ello, es el mismo procedimiento ejecutado en el punto **4.5.1.1 Pruebas en Entorno Privado** a diferencia que se establece la dirección IP pública 186.28.115.78 del objeto IoT como se muestra en la Ilustración 93.

Ilustración 92. Creación de Escaneo IoT ESP32

Fuente: Elaboración Propia

- 3) Se inicia proceso de descubrimiento de dispositivo IoT de acuerdo a los parámetros establecidos – Ilustración 94.

Ilustración 93. Inicio de Proceso de Descubrimiento

Fuente: Elaboración Propia

- 4) Teniendo como resultado un 100% en el proceso de descubrimiento del dispositivo IoT ESP32 como se muestra en la Ilustración 95, se puede detallar información del elemento encontrado un resultado, teniendo en cuenta que se hace el escaneo por medio de un router y se tiene un filtro adicional por lo que la información puede variar a la prueba anterior en entorno privado.

Ilustración 94. Resultado Proceso de Escaneo ESP32

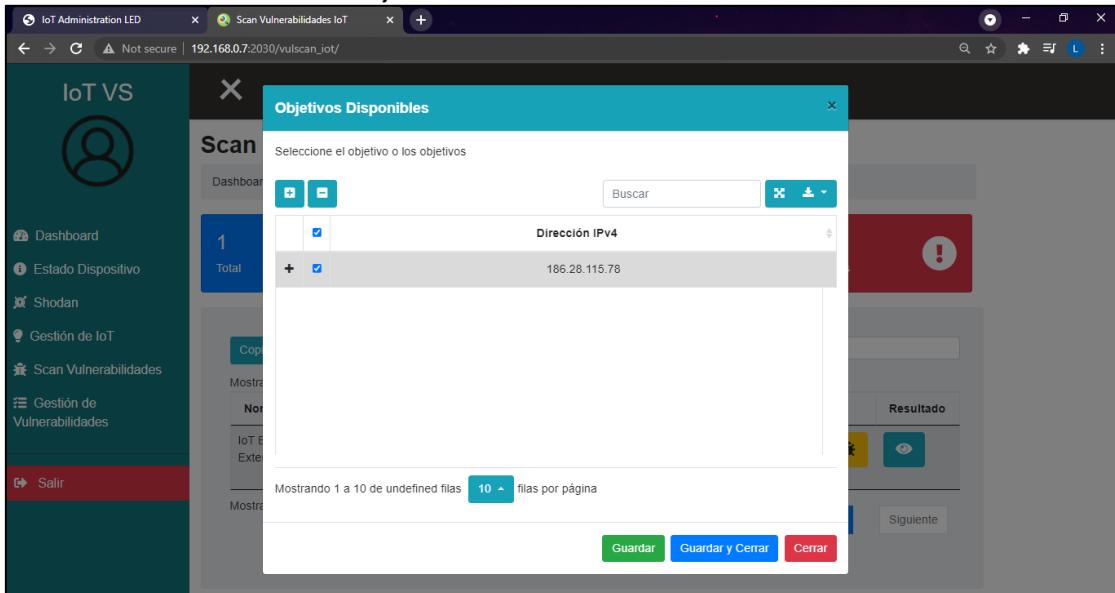
Puerto	Protocolo	Estado	Nombre Servicio	Versión Servicio
TCP/23	telnet	filtered		
TCP/80	tcpwrapped	open		
TCP/443	https	open		

Fuente: Elaboración Propia

- 5) Teniendo presente parámetros propios de la tarjeta IoT ESP32 descubiertos por el esquema de software implementado en la Raspberry Pi, ahora en el apartado denominado "Scan Vulnerabilidades" se procede en primera instancia a seleccionar

el objetivo como se muestra en la Ilustración 96, que para esta prueba solo se encuentra disponible la tarjeta ESP32.

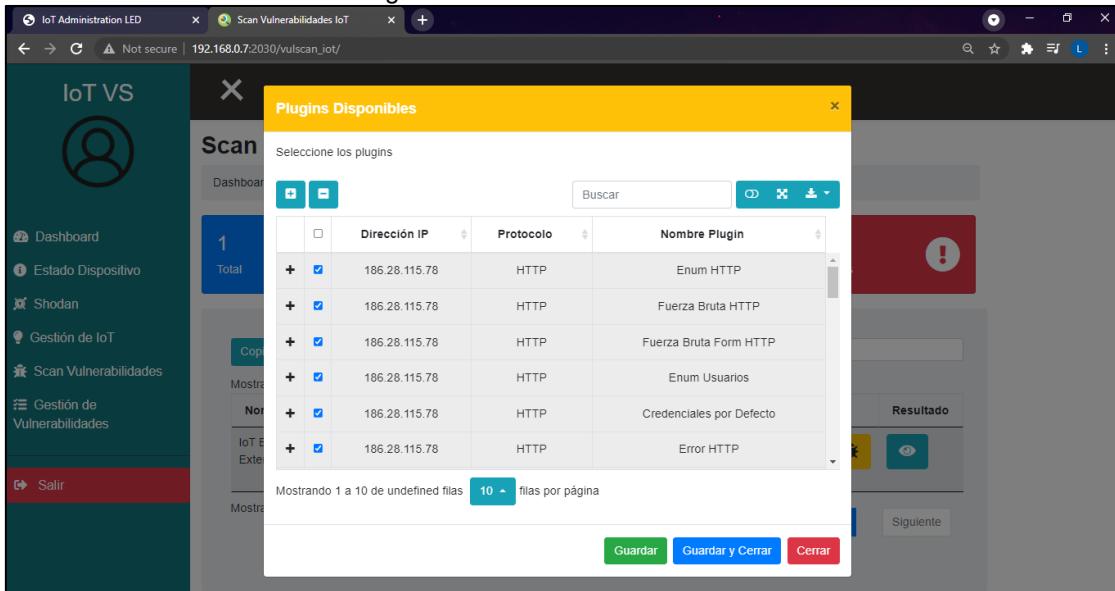
Ilustración 95. Selección de Objetivo Para Detección de Vulnerabilidades



Fuente: Elaboración Propia

- 6) Descubierto por módulo “Gestión de IoT” el puerto TCP/80 correspondiente a HTTP en la tarjeta IoT ESP32 el software de forma automática muestra plugins correspondientes a este servicio, que permitirán hacer el descubrimiento de vulnerabilidades. Para efectos de prueba se seleccionarán todos los plugins disponibles como se evidencia en la Ilustración 97.

Ilustración 96. Selección de Plugins Para Descubrimiento de Vulnerabilidades



Fuente: Elaboración Propia

- 7) Se inicia proceso de detección de vulnerabilidades en dispositivo IoT de acuerdo a los parámetros establecidos – Ilustración 98.

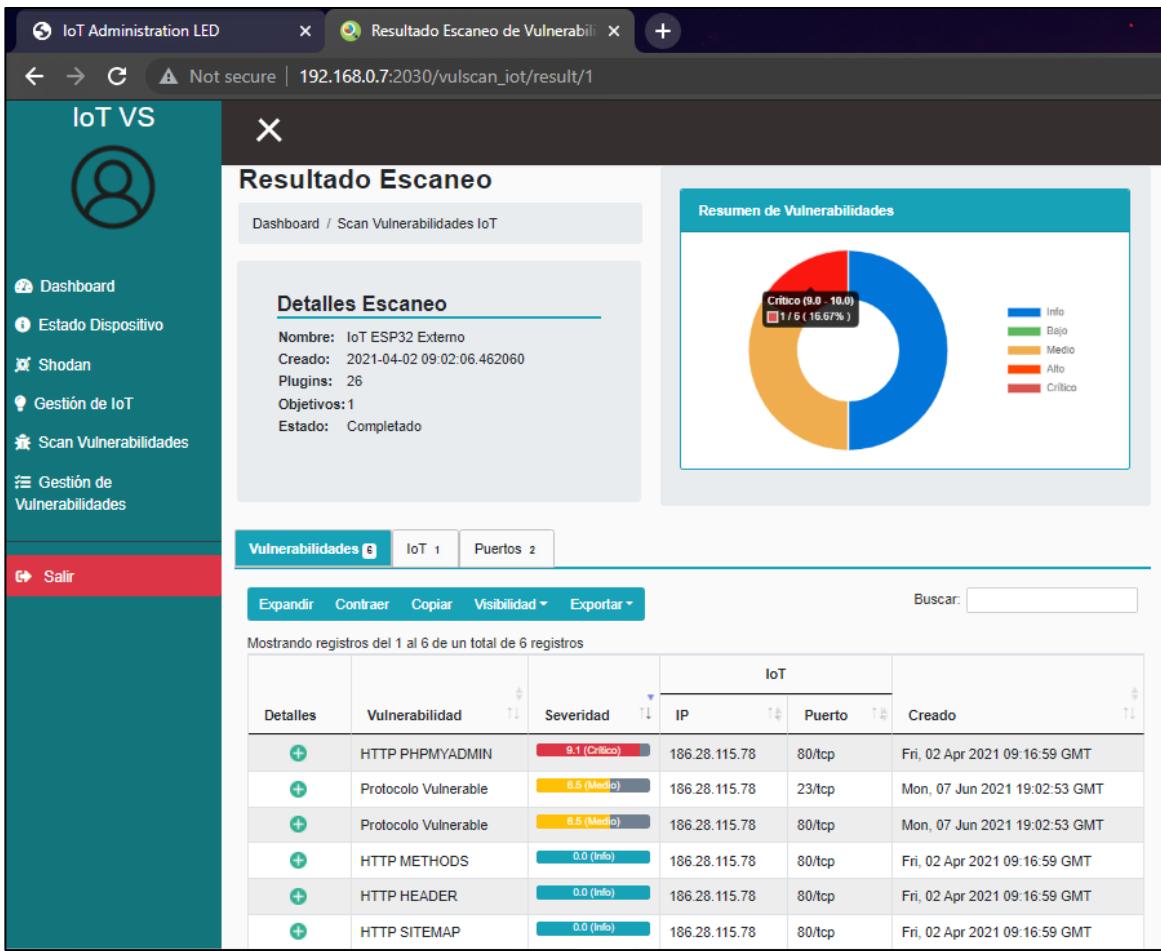
Ilustración 97. Inicio Proceso de Detección de Vulnerabilidades

The screenshot shows a web-based interface titled 'Scan Vulnerabilidades IoT'. On the left, there is a sidebar with a user icon and several menu items: Dashboard, Estado Dispositivo, Shodan, Gestión de IoT, Scan Vulnerabilidades, Gestión de Vulnerabilidades, and Salir. The main area is titled 'Scan Vulnerabilidades IoT' and displays a summary of the scan status: Total 1, Terminados 0, Pendientes 1, and Errores 0. Below this, a table lists one record: IoT ESP32 Externo, Control de Bombillo, 186.28.115.78, Fri, 02 Apr 2021 09:02:06 GMT, and a progress bar at 0%. A green toast notification at the bottom right says 'Proceso Hecho' and 'IoT ESP32 Externo - Ha comenzado con éxito'. The table has columns: Nombre, Descripción, Objetivo(s), Fecha/Hora, Estado, Acciones, and Resultado.

Fuente: Elaboración Propia

Resultados

En el resultado obtenido de la prueba automatizada de ciber-ataque sobre la tarjeta IoT ESP32 de forma externa se encuentra el uso de 26 Plugins, en donde se detectaron 6 hallazgos que pueden ayudar a identificar problemas a nivel seguridad de software sobre este dispositivo – como se muestra en la Ilustración 99 – teniendo en cuenta que ha identificado el puerto TCP/23 como un puerto vulnerable.

Ilustración 98. Resultado Escaneo de Vulnerabilidades

Fuente: Elaboración Propia

El conglomerado completo cuenta con hallazgos de nivel informativos, medios y críticos, cada uno identificado de forma clara, con su respectiva descripción, resultado e información que ayuda al usuario final a la toma de decisiones y/o proceso adecuado de remediación sobre la vulnerabilidad como se muestra en la Ilustración 100.

Ilustración 99. Resumen Hallazgos Encontrados

The screenshot shows a web browser window titled "Resultado Escaneo de Vulnerabilidad" with the URL "192.168.0.7:2030/vulscan_iot/result/1". The page displays a summary of findings:

- Protocolo Vulnerable:** Protocolo tcp/80 inseguro
- Resumen:** Servicios de Red Inseguros
- Resultado de Vulnerabilidad:** Protocolo tcp/80 inseguro
- Remediación de Vulnerabilidad:** Se recomienda asegurar los servicios de red de acuerdo a: 1. Asegurar que solo los puertos necesarios estén expuestos y disponibles 2. Hacer uso de protocolos seguros como SSH 3. Garantizar que los servicios no sean vulnerables al desbordamiento del búfer y a los ataques de fuzzing. 4. Garantizar que los servicios no sean vulnerables a los ataques DoS. 5. Asegurar que los puertos o servicios de red no estén expuestos a Internet a través de UPnP, por ejemplo.

Below the summary, there are two tables:

	HTTP METHODS	0.0 (Info)	186.28.115.78	80/tcp	Fri, 02 Apr 2021 09:16:59 GMT
+	HTTP HEADER	0.0 (Info)	186.28.115.78	80/tcp	Fri, 02 Apr 2021 09:16:59 GMT

	Resumen:
+	Informativo

Fuente: Elaboración Propia

Hallazgos

El nivel de hallazgos informativos, bajos, medios, altos y críticos generados por el software es el mismo al obtenido en las pruebas realizadas en el apartado **4.5.1.1 Pruebas en Entorno Privado**, teniendo como diferencial el entorno público de ejecución y alcance del objeto IoT que en este caso es la tarjeta ESP32, en donde el tráfico en entrante/saliente viajando por internet, demostrando la usabilidad del esquema de software no solo de forma interna sino su alcance a redes remotas ubicadas en ambientes geográficos diferentes.

Contemplando un nivel de amenaza para el objeto IoT ESP32 que se muestra en la siguiente Tabla 65.

Tabla 65. Muestra Amenaza ESP32

Riesgo	Porcentaje	Número de Vulnerabilidades
Critico	16.67%	1
Alto	0.0%	0
Medio	33.33%	2
Bajo	0.0%	0
Informativo	50.00%	3
	100%	6

Fuente: Elaboración Propia

OBSERVACIONES

La velocidad en el descubrimiento y detección de vulnerabilidades en el objetivo es directamente proporcional a la tarjeta Raspberry Pi y latencia de comunicación que se encuentre entre el origen y destino de la comunicación.

4.5.2 Desarrollo Fase II de Pruebas

4.5.2.1 Pruebas Unitarias

Para iniciar esta segunda fase de pruebas, se opta por hacer uso de las pruebas unitarias bajo el modelo de caja negra, validando de forma independiente cada módulo y que se encuentre en un funcionamiento acorde a lo necesario y especificado en la fase de desarrollo.

A continuación, en la Tabla 66 se observa el resultado de las pruebas unitarias sobre la autenticación.

Tabla 66. Pruebas Unitarias de Autenticación

Módulo	Autenticación
Estado Aprobación	Aceptado
Versión	v1
Entrada	Credenciales de usuario
Resultado Obtenido	Ingreso a módulo principal de dashboard
Resultado Esperado	El comportamiento generado fue el esperado y aceptado por el usuario final

Fuente: Elaboración Propia

La Tabla 67 muestra el resultado de las pruebas unitarias sobre el módulo estado dispositivo.

Tabla 67. Pruebas Unitarias Módulo Estado Dispositivo

Módulo	Estado Dispositivo
Estado Aprobación	Aceptado
Versión	v1
Entrada	N/A
Resultado Obtenido	Muestra información a nivel de red de la Raspberry Pi
Resultado Esperado	El comportamiento generado fue el esperado y aceptado por el usuario final

Fuente: Elaboración Propia

La Tabla 68 muestra el resultado de las pruebas unitarias sobre el módulo Shodan.

Tabla 68. Pruebas Unitarias Módulo Shodan

Módulo	Shodan
Estado Aprobación	Aceptado
Versión	v1
Entrada	Parámetro de búsqueda
Resultado Obtenido	Muestra resultados relevantes al parámetro especificado por el usuario
Resultado Esperado	El comportamiento generado fue el esperado y aceptado por el usuario final

Fuente: Elaboración Propia

La Tabla 69 muestra el resultado de las pruebas unitarias sobre el módulo gestión de IoT.

Tabla 69. Pruebas Unitarias Módulo Gestión de IoT

Módulo	Gestión de IoT
Estado Aprobación	Aceptado
Versión	v1
Entrada	<ul style="list-style-type: none"> - Nombre - Descripción - Dirección IPv4 de objeto IoT
Resultado Obtenido	<p>Creación de escaneo de forma exitosa, modificándolo, iniciándolo y eliminándolo acorde a lo requerido.</p> <p>Resultado ordenado de objetos IoT encontrados</p>
Resultado Esperado	El comportamiento generado fue el esperado y aceptado por el usuario final

Fuente: Elaboración Propia

La Tabla 70 muestra el resultado de las pruebas unitarias sobre el módulo scan vulnerabilidades.

Tabla 70. Pruebas Unitarias Módulo Scan Vulnerabilidades

Módulo	Scan Vulnerabilidades
Estado Aprobación	Aceptado
Versión	v1
Entrada	<p>Selección de elementos:</p> <ul style="list-style-type: none"> - Objeto IoT - Plugins
Resultado Obtenido	Ejecución correcta del escaneo de vulnerabilidades y resultados ordenados/acordes a lo encontrado
Resultado Esperado	El comportamiento generado fue el esperado y aceptado por el usuario final

Fuente: Elaboración Propia

La Tabla 71 muestra el resultado de las pruebas unitarias sobre el módulo gestión de vulnerabilidades.

Tabla 71. Pruebas Unitarias Módulo Gestión de IoT

Módulo	Gestión de Vulnerabilidades
Estado Aprobación	Aceptado
Versión	v1
Entrada	<ul style="list-style-type: none"> - Nombre Escaneo - Selección de Vulnerabilidad - Titulo Actividad - Estado

	<ul style="list-style-type: none"> - Fecha/Hora Inicio - Fecha/Hora Fin
Resultado Obtenido	Creación de actividad de remediación de forma exitosa, modificándolo y eliminándolo acorde a lo requerido.
Resultado Esperado	El comportamiento generado fue el esperado y aceptado por el usuario final

Fuente: Elaboración Propia

La Tabla 72 muestra el resultado de las pruebas unitarias sobre el módulo dashboard

Tabla 72. Pruebas Unitarias Módulo Dashboard

Módulo	Dashboard
Estado Aprobación	Aceptado
Versión	v1
Entrada	N/A
Resultado Obtenido	Muestra por medio de gráficas estado global de vulnerabilidades, nivel de amenaza al que se encuentra expuestos puertos y objetos IoT.
Resultado Esperado	El comportamiento generado fue el esperado y aceptado por el usuario final

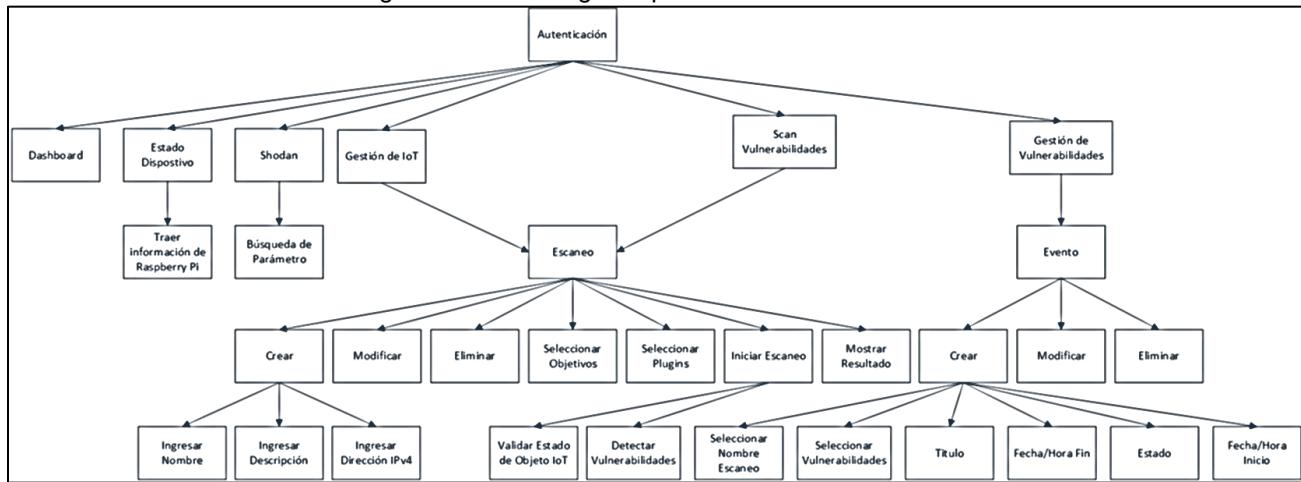
Fuente: Elaboración Propia

4.5.2.2 Pruebas de Humo

De forma consecutiva para esta fase de pruebas, se opta por hacer uso del modelo denominado prueba de humo o también “smoke test”, con el cual se tuvo un panorama general y rápido sobre los aspectos funcionales del software, sin entrar en detalles técnicos en la búsqueda de bugs sino para asegurar que el funcionamiento general básico se encuentre estable y responda al comportamiento esperado. Detectando errores tempranos en realeases y asegurando resultados en las pruebas unitarias.

4.5.2.3 Pruebas de Integración

La ejecución exitosa de las pruebas de integración, permitieron verificar los componentes del software ensamblados correctamente, una vez han sido probados de forma unitaria en las interacciones de SCRUM del proceso de codificación, con el objetivo de corroborar una interacción correcta por medio de la interfaz gráfica mostrada al usuario final, evitando faltas de coherencia entre lo que se espera de un módulo y lo que en realidad es mostrado. Se siguió el proceso de ejecución que se puede observar en la Ilustración 101, haciendo uso de la estrategia “Top Down”.

Ilustración 100. Prueba de Integración - Estrategia Top Down

Fuente: Elaboración Propia

4.5.2.4 Pruebas de Funcionalidad

Aplicando pruebas de funcionalidad sobre la herramienta de software desarrollada, se comprobaron las características críticas del sistema y que se satisfacen las especificaciones funcionales planteadas desde el diseño, garantizando que propiedades básicas se comporten según lo esperado sin ningún tipo de problema relevante o de categoría error grave, comprometiendo su uso normal.

4.5.2.5 Pruebas de Interfaz

Tras haber navegado por los objetos de la herramienta de software, se refleja funcionalidad y requisitos fundamentales en óptimo estado, a través del manejo de interfaz gráfica de menús, movimientos del mouse, posiciones de objetos, estados, botones, etc., confirmando una interacción del usuario fluida, clara e intuitiva.

4.5.3 Resultados

4.5.3.1 Resumen de Resultados

Cumpliendo satisfactoriamente el plan de pruebas programadas para el software desarrollado, se muestra a continuación la Tabla 73 que ayuda a entender de forma global el resultado final, determinando si el software cumple o no con los requerimientos del producto necesario para este proyecto.

Tabla 73. Resumen de Pruebas

Nombre de Prueba	Criterio de Aprobación				Requerimientos	
	Aceptado	Errores Leves	Errores Medios	Errores Graves	SI Cumple	NO Cumple
Pruebas en Entorno Privado	X				X	
Pruebas en Entorno Público	X				X	

Pruebas Unitarias	X				X	
Pruebas de Humo	X				X	
Pruebas de Integración	X				X	
Pruebas de Funcionalidad	X				X	
Prueba de Interfaz	X				X	

Fuente: Elaboración Propia

4.5.3.2 Análisis de Resultados

Realizando un proceso de análisis frente a los resultados obtenidos en cada una de las pruebas, se puede observar un cumplimiento conforme a los requerimientos estipulados para la culminación exitosa del presente proyecto, considerando que no se presentaron errores que puedan afectar de alguna manera el comportamiento del software u obtener resultados no previstos. Con la finalidad que sea usado como herramienta para la toma de decisiones y aprovechamiento de resultados confiable, gracias a la recopilación, procesamiento y exposición de información de valor.

Por otra parte, es importante resalta que actores externos ajenos al software pueden influir en el flujo y rendimiento de la ejecución; tales como el procesador, espacio disponible en disco, cantidad de memoria RAM, conexión a internet, etc., que a pesar de elegirse para este proyecto implementarse con Raspberry Pi 4 modelo B, la gran variedad de placas de tamaño reducido existentes en el mercado y la flexibilidad con el que fue desarrollado el software, permiten un despliegue en cualquiera de placa de tamaño reducido que cuenta con sistema operativo basado el GNU/Linux.

5. CONCLUSIONES Y TRABAJOS FUTUROS

5.1 CONCLUSIONES

Tras la culminación satisfactoria del presente proyecto, se ha logrado realizar la creación de un prototipo de software funcional implementado en Raspberry Pi actuando como un dispositivo de borde, en donde se lograron detectar vulnerabilidades presentes en objetos IoT, haciendo uso de proyectos base como el OWASP Internet of Things y el IoT Security Compliance Framework, que ayudaron al modelamiento e identificación de vulnerabilidades presentes en dispositivos IoT, convirtiéndose en amenazas activas en diferentes ámbitos del mundo global.

Aplicando un proceso secuencial de pasos aplicados al software desarrollado, se puede concluir que no solo es funcional para la detección y clasificación de vulnerabilidad, sino que también permite hacer un seguimiento completo al ciclo de vida de una vulnerabilidad que se pueda presentar en un objeto IoT, abordando desde el proceso de descubrimiento hasta el seguimiento adecuado para una remediación efectiva. Mostrando que es aplicable tanto en entornos externos como internos, brindando a un usuario final del software la posibilidad de tener un panorama general de una infraestructura tecnológica conformada con objetos IoT.

De forma masiva se distribuyen objetos IoT sin contar con un desarrollo seguro de software, dejando de lado aspecto fundamentales de seguridad que pueden ser aprovechados por actores maliciosos. Adicionalmente, deficiencias en la implementación y mantenimiento de objetos IoT pueden favorecer a accesos no autorizados, todo esto mostrado en las pruebas controladas de ataque sobre la tarjeta IoT ESP32, evidenciando una muestra de lo vulnerables que pueden llegar a convertirse estos dispositivos sin un debido control en fases iniciales hasta fases de manteniendo, convirtiéndose en una amenaza que puede ayudar a comprometer una infraestructura tecnológica de una organización.

5.2 TRABAJOS FUTUROS

Considerando el nivel aplicación del software desarrollado, se contempla a futuro hacer un proceso de maduración y acoplamiento a frameworks web de alto nivel para un proceso de distribución de forma masiva, el cual sea usado por particulares y organizaciones para el desempeño efectivo de control sobre sus activos informáticos. Así mismo, se contempla realizar pruebas en entornos productivos y sobre diversos objetos IoT para medir el rendimiento general del software, con el objetivo de realizar mejoras en resultados, acoplamiento a mayores protocolos presentes en IoT y determinar que otros aspectos carentes de seguridad puede abordar la herramienta.

6. REFERENCIAS BIBLIOGRÁFICAS

- 3Ciencias. (Octubre de 2018). *3Ciencias*. Recuperado el 08 de Abril de 2020, de <https://bit.ly/3IDNVS7>
- Abawajy, J., Shamsul, H., Shaila, S., Mohammad, M., & Ahmad, A. (2018). ScienceDirect. En *Future Generation Computer Systems* (págs. 525-538). ELSEVIER. Obtenido de <https://bit.ly/3p0mHYb>
- Agency, D. A. (Septiembre de 1981). *tools.ietf.org*. Recuperado el 10 de Enero de 2021, de <https://bit.ly/3ax369w>
- Agency, D. A. (Septiembre de 1981). *tools.ietf.org*. Recuperado el 10 de Enero de 2021, de <https://bit.ly/3FI5Cbb>
- Akamai. (2016). *Akamai*. Recuperado el 4 de Mayo de 2020, de <https://bit.ly/3v9oQSc>
- Alarcón, V., González, C., & González, A. (7 de Julio de 2017). *Protocolo UPnP*. Recuperado el 7 de Mayo de 2020, de <https://bit.ly/3BFxryH>
- Alcaraz, M. (Octubre de 2014). *Universidad Católica*. Recuperado el 19 de Mayo de 2020, de <https://bit.ly/3DG5dUO>
- Alliance, Z. (s.f.). *Zigbee Alliance*. Recuperado el 7 de Mayo de 2020, de <https://bit.ly/2YPCEFG>
- Alonso, C. (14 de Noviembre de 2018). *Un Informático en el Lado del Mal*. Recuperado el 14 de Mayo de 2020, de <https://bit.ly/3p5Llqp>
- Álvaro Núñez-Romero Casado, J. J. (Mayo de 21 de 2016). *SlideShare*. Recuperado el 1 de Abril de 2020, de <https://bit.ly/3BFxTwT>
- AndalucíaCERT. (28 de Julio de 2014). *Seguridad Andalucía*. Recuperado el 27 de Abril de 2020, de <https://bit.ly/2YIXlmo>
- Andrés, M. B. (2018). *Editorial EUS*. Recuperado el 19 de Mayo de 2020, de <https://bit.ly/3p1bR4e>
- Antonio, R. (25 de Mayo de 2015). *Globb Security*. Obtenido de <https://bit.ly/3BJAVQz>
- ARDUINO. (2020). *ARDUINO*. Recuperado el 14 de Mayo de 2020, de <https://bit.ly/3mT36pT>
- Ariganello, E. (2014). Redes Cisco. En *Guía de estudio para la certificación CCNA Security* (págs. 47-48). Madrid: Ra-Ma.
- Bankinter, F. d. (2011). Recuperado el 19 de Mayo de 2020, de <https://bit.ly/30fTHBa>
- Baquero Rey, L., & Hernández Bejarano, M. (2018). *Las Tecnologías de la Información y la Comunicación y su Aplicación Empresarial*. Bogotá: Scientometrics e Researching Consulting Group.
- Benlgadima. (26 de Octubre de 2016). *imperva*. Recuperado el 4 de Mayo de 2020, de <https://bit.ly/3AFvLUz>
- Bevan, N. (1997). *ACADEMIA*. Recuperado el 07 de 04 de 2020, de <https://bit.ly/3FMFQCT>

- Blanco, R., Fontrodona, J., & Poveda, C. (s.f.). *Mincultura España*. Recuperado el 18 de Mayo de 2020, de <https://bit.ly/3FUYNDx>
- Burgett, A. (10 de Enero de 2019). *KirkpatrickPrice*. Recuperado el 28 de Abril de 2020, de <https://bit.ly/3aBIs9k>
- Calvo Ortega, G., & García Valdés, Á. (Enero de 2018). Recuperado el 4 de Mayo de 2020, de <https://bit.ly/3BETR33>
- Cámara Nebreda, J. (2017). *RIUBU*. Recuperado el 15 de Mayo de 2020, de <https://bit.ly/3aztRKe>
- Carretero Aguilar, C., Añón Rodríguez, M., Galán Obregón, A., Riol, J. M., García Vázquez, A., & Peña Mújica, G. (13 de Febrero de 2019). *Esmartcity*. Recuperado el 29 de Abril de 2020, de <https://bit.ly/3ayljdP>
- Castresana Sáenz, C. (2016). *Bibliote Universidad de La Rioja*. Recuperado el 18 de Mayo de 2020, de <https://bit.ly/3iXLnMQ>
- Cendon, B. (16 de Enero de 2017). *Bruno Cendon*. Recuperado el 08 de Abril de 2020, de <https://bit.ly/3FMaQTu>
- Cera Cárdenas, J., Martínez Otero, L., Rojas Blandón, J., Villaveces Santander, J., & Sanmartín Mendoza, P. (2015). *Joseph Cera Cárdenas*. Recuperado el 19 de Mayo de 2020, de <https://bit.ly/2YOiY4D>
- Ciberseguridad, I. N. (2017). *INCIBE*. Recuperado el 30 de Junio de 2021, de <https://bit.ly/2YMw96O>
- CLOUDFLARE. (14 de Diciembre de 2017). *CLOUDFLARE*. Recuperado el 4 de Mayo de 2020, de <https://bit.ly/2YPbkqE>
- Colombia, C. d. (05 de Enero de 2009). *Alcaldía de Bogotá*. Recuperado el 30 de Mayo de 2021, de <https://bit.ly/3DByaBq>
- David A. Franco, J. L. (3 de Enero de 2013). *SCIELO*. Recuperado el 28 de Abril de 2020, de <https://bit.ly/3BJBqtV>
- del Val Román, . L. (Octubre de 2016). *CODDII ORG*. Recuperado el 18 de Mayo de 2020, de <https://bit.ly/3AC2Mkh>
- Dennis, A. K. (2016). *Raspberry Pi Computer Architecture Essentials*. Packt Publishing.
- Didiana Velásquez, L. (2019). *MinTIC Colombia*. Recuperado el 18 de Mayo de 2020, de <https://bit.ly/3mUDYz6>
- Duran Caastillo, E. (2019). Recuperado el 19 de Mayo de 2020, de <https://bit.ly/3BApYAN>
- EC-Council. (20 de Noviembre de 2018). *EC-Council*. Recuperado el 28 de Abril de 2020, de <https://bit.ly/3vceSiR>
- Education, I. C. (19 de Agosto de 2020). *IBM* . Obtido de <https://www.ibm.com/cloud/learn/api>
- Education, I. C. (6 de Abril de 2021). *IBM*. Obtido de <https://ibm.co/3o8SKV0>
- Elliot, R. (22 de Octubre de 2018). *Electro Maker*. Recuperado el 13 de Mayo de 2020, de <https://bit.ly/3AEF7Qi>

- ESET. (4 de Agosto de 2014). *ESET*. Recuperado el 11 de Enero de 2021, de <https://bit.ly/3BHNVGe>
- Española, R. A. (s.f.). *RAE*. Recuperado el 08 de Abril de 2020, de <https://bit.ly/2YJeRH9>
- Exposures, C. C. (1 de Febrero de 2020). *CVE Common Vulnerabilities and Exposures*. Recuperado el 22 de Abril de 2020, de <https://bit.ly/3vnIXxv>
- Foundation, T. I. (23 de Septiembre de 2015). *The Internet of Things Security Foundation*. Recuperado el 11 de Enero de 2021, de <https://bit.ly/30oZi8u>
- Foundation, T. I. (Mayo de 2020). *The Internet of Things Security Foundation*. Recuperado el 11 de Enero de 2021, de <https://bit.ly/3FI6oVD>
- FRAMEWORK, C. (5 de Febrero de 2018). *NIST CYBERSECURITY FRAMEWORK*. Recuperado el 9 de Enero de 2021, de <https://bit.ly/3mVcOs3>
- Fundation, I. S. (2015). Recuperado el 5 de Mayo de 2020, de <https://bit.ly/2YPbFcU>
- Fundation, I. S. (Diciembre de 2016). Recuperado el 5 de Mayo de 2020, de <https://bit.ly/3FD159Y>
- García Cobo, J. (s.f.). *Hardware Libre*. Recuperado el 14 de Mayo de 2020, de <https://bit.ly/3DCqoXY>
- Garcia Rambla, J. L., Alonso, C., & González, P. (2017). En *Ataques en Redes de Datos IPV4 e IPV6* (pág. 17). Madrid: 0xWORD.
- García, D. (9 de Octubre de 2018). *ElevenPaths*. Recuperado el 4 de Mayo de 2020, de <https://bit.ly/3BAqezL>
- Grizhnevich, A. (1 de Abril de 2018). *ScienceSoft*. Recuperado el 20 de Mayo de 2020, de <https://bit.ly/3FEqwYV>
- Hallberg, B. A. (2007). Protocolo TCP/IP. En *Fundamentos de Redes* (págs. 95 - 96). McGraw-Hill.
- Hosmer, C. (2018). En *Defending IoT Infraestructures with the Raspberry Pi* (pág. 6). South Carolina: APRESS.
- Hosmer, C. (2018). En *Defending IoT Infraestructures with the Raspberry Pi* (pág. 4). South Carolina: APRESS.
- Hruska, J. (11 de Abril de 2017). (ExtremeTech) Recuperado el 4 de Mayo de 2020, de <https://bit.ly/3DHFOtQ>
- ICANNWiki. (2 de Noviembre de 2019). *ICANNWiki*. Recuperado el 4 de Mayo de 2020, de <https://bit.ly/3ALc8dO>
- INCIBE. (20 de Marzo de 2017). *INCIBE*. Recuperado el 08 de Abril de 2020, de <https://bit.ly/3aCSvd0>
- Insurgentes, U. (s.f.). *Universidad Insurgentes*. Recuperado el 10 de Enero de 2021, de <https://bit.ly/3ayQn67>
- INTEF. (s.f.). *INTEF*. Recuperado el 08 de Abril de 2020, de <https://bit.ly/3mVdcH1>
- IoT, O. (1 de Noviembre de 2019). Recuperado el 5 de Mayo de 2020, de <https://bit.ly/3lFVqZ2>

- IoTGoat, O. (30 de Marzo de 2020). *Github OWASP IoTGoat*. Recuperado el 5 de Mayo de 2020, de <https://bit.ly/3BMg8fq>
- ISECOM. (14 de Diciembre de 2010). *ISECOM*. Recuperado el 9 de Enero de 2021, de <https://bit.ly/3DyPOWg>
- ISO. (1994). *ISO 5725*. Recuperado el 12 de Enero de 2021, de <https://bit.ly/3vbCdl5>
- JUAN CARLOS PÉREZ NAVA, E. R. (s.f.). *UNAM MX*. Recuperado el 2 de Junio de 2021, de <https://bit.ly/3BHOxvw>
- Karen Rose, S. E. (Octubre de 2015). *Interner Society*. Recuperado el 1 de Abril de 2020, de <https://bit.ly/3iWfmon>
- Kaspersky. (11 de Septiembre de 2014). Recuperado el 14 de Abril de 2020, de <https://bit.ly/3mT0RTK>
- Ken Schwaber, J. S. (Julio de 2016). *Scrumguides*. Recuperado el 26 de Junio de 2021, de <https://bit.ly/3mQ4OII>
- KirstenS, J. M. (s.f.). *OWASP Cross Site Scripting (XSS)*. Recuperado el 2020 de Abril de 2020, de <https://bit.ly/3FK3lw5>
- Kleinman, Z. (6 de Agosto de 2013). Recuperado el 04 de Abril de 2020, de <https://bbc.in/3BEPTHr>
- Kniberg, H. (2007). *SCRUM Y XP DESDE LAS TRINCHERAS*. Estados Unidos de América: C4Media Inc.
- Kochetkova, K. (Octubre de 26 de 2016). *Kaspersky*. Recuperado el 4 de Mayo de 2020, de <https://bit.ly/3vbEHjh>
- Kyriakos Kritikos, K. M. (Diciembre de 2019). *ELSEVIER*. Recuperado el 27 de Abril de 2020, de <https://bit.ly/3j2ta0L>
- LACNIC. (Agosto de 2020). *LACNIC*. Recuperado el 8 de Noviembre de 2020, de <https://bit.ly/3DG6NGe>
- Lakhani, A., & Muniz, J. (2015). En *Penetration Testing with Raspberry Pi* (págs. 31-32). Birmingham: Pack Publishing.
- LASTKOW, S. (27 de Enero de 2017). *Atlas Obscura*. Recuperado el 28 de Abril de 2020, de <https://bit.ly/3BG5Flw>
- Libertadores, F. U. (24 de Junio de 2019). *Fundación Universitaria Los Libertadores*. Recuperado el 10 de Junio de 2021, de <https://bit.ly/3mROk2P>
- López, A. (21 de Julio de 2015). *Incibe-Cert*. Recuperado el 27 de Abril de 2020, de <https://bit.ly/3AC492p>
- Magazine, U. N. (05 de Marzo de 2019). *UseC Network Magazine*. Recuperado el 07 de Abril de 2020, de <https://bit.ly/3BGBwma>
- Maksimović, M. (28 de Agosto de 2017). *Scielo*. Recuperado el 19 de Mayo de 2020, de <https://bit.ly/2YWjZYO>
- Manager, S. (2015). *Scrum Manager*. Recuperado el 26 de Junio de 2021, de <https://bit.ly/3iZafnM>

- Martínez, R. (8 de Abril de 2016). *NOTICIAS DE SEGURIDAD INFORMATICA*. Recuperado el 21 de Abril de 2020, de <https://bit.ly/3mVdDB9>
- Mendez, L. (17 de Febrero de 2017). *Cisco Latinoameérica*. Recuperado el 29 de Abril de 2020, de <https://bit.ly/3DTFIzN>
- Microsoft. (15 de Agosto de 2015). Recuperado el 08 de Abril de 2020, de <https://bit.ly/3IFVTKM>
- Mitre, C. (s.f.). *CVE Mitre*. Recuperado el 22 de Abril de 2020, de <https://bit.ly/3p0laBp>
- Moreno Saiz, S. (2915). *Universidad Politécnica de Madrid*. Recuperado el 20 de Mayo de 2020, de <https://bit.ly/2XfC0AM>
- Morgan, S. (18 de Julio de 2019). *Cybersecurity Centres*. Recuperado el 07 de Abril de 2020, de <https://bit.ly/3mQ3tSm>
- Morteza Verdi, A. S. (3 de Octubre de 2019). Recuperado el 16 de Abril de 2020, de <https://bit.ly/3mVe26D>
- Navarro Cadavid, A., Fernández Martínez, J., & Morales Vélez, J. (20 de Septiembre de 2013). *Universidad Autónoma del Caribe*. Recuperado el 20 de Mayo de 2020, de <https://bit.ly/3IEaeaw>
- Néstor Dabío Duque Méndez, A. T. (s.f.). *UNAL*. Recuperado el 28 de Abril de 2020, de <https://bit.ly/3v8KWEz>
- Nilsson, J., & Virta, V. (2006). Recuperado el 28 de Abril de 2020, de <https://bit.ly/3mOrKYZ>
- NIST. (s.f.). *NIST*. Recuperado el 27 de Abril de 2020, de <https://bit.ly/3DIKEaz>
- ODROID. (4 de Abril de 2020). *Wiki ODROID*. Recuperado el 13 de Mayo de 2020, de <https://bit.ly/2Xe9z6e>
- ODROID. (s.f.). *ODROID*. Recuperado el 13 de Mayo de 2020, de <https://bit.ly/3axUzmE>
- Org, F. (2015). *First Org*. Recuperado el 11 de Enero de 2021, de <https://bit.ly/3vcfHZm>
- ORG, F. (s.f.). *FIRST ORG*. Recuperado el 27 de Abril de 2020, de <https://www.first.org/members/map>
- ORG, F. (s.f.). *FIRST ORG*. Recuperado el 27 de Abril de 2020, de <https://bit.ly/2X8KFVq>
- ORG, O. (1 de Diciembre de 2001). *OWASP ORG*. Recuperado el 11 de Enero de 2021, de <https://bit.ly/3BHPxQi>
- ORG, O. (2004). *OWASP Web Security Testing Guide*. Recuperado el 11 de Enero de 2021, de <https://bit.ly/3aBxi3i>
- ORG, O. (2013). *OWASP Top Ten*. Recuperado el 11 de Enero de 2021, de <https://bit.ly/3DEkfup>
- ORG, O. (9 de Septiembre de 2018). *OpenWrt ORG*. Recuperado el 14 de Mayo de 2020, de <https://bit.ly/3BHesUb>
- ORG, O. (2018). *OWASP Internet of Things*. Recuperado el 11 de Enero de 2021, de <https://bit.ly/3mQi007>
- OS, B. P. (21 de Abril de 2020). *Banana Pi OS*. Recuperado el 13 de Mayo de 2020, de <https://bit.ly/3j0o9Wy>

- Oviedo, U. d. (s.f.). *Universidad de Oviedo*. Recuperado el 10 de Enero de 2021, de <https://bit.ly/3oYRNPE>
- OWASP. (s.f.). Recuperado el 21 de Abril de 2020, de <https://bit.ly/3j0HWVU>
- OWASP. (2018). OWASP. Recuperado el 07 de Abril de 2020, de <https://bit.ly/3aAuah>
- OWASP. (2018). *OWASP Top 10 IoT*. Recuperado el 9 de Enero de 2021, de <https://bit.ly/3DFwuqE>
- OWASP. (s.f.). *OWASP Buffer Overflow Vulnerabilities*. Recuperado el 27 de Abril de 2020, de <https://bit.ly/3DBejSQ>
- OWASP. (s.f.). *OWASP SQL Injection*. Recuperado el 27 de Abril de 2020, de <https://bit.ly/3iXNvnO>
- OWASP. (s.f.). *OWASP TESTING GUIDE*. Recuperado el 27 de Abril de 2020, de <https://bit.ly/3va6YXj>
- Packard, H. (29 de Julio de 2014). *Hewlett Packard*. Recuperado el 07 de Abril de 2020, de <https://bit.ly/3FMHcNK>
- Palacio, M. (Junio de 2021). *Scrummanager*. Recuperado el 26 de Junio de 2021, de <https://bit.ly/3lBFmr6>
- Pentest-Standard. (16 de Agosto de 2014). *Pentest-Standard*. Recuperado el 9 de Enero de 2021, de <https://bit.ly/30jbo2J>
- Perez, M. A. (8 de Julio de 2014). *Blog Think Big*. Recuperado el 13 de Mayo de 2020, de <https://bit.ly/3DyR1wM>
- Pi, B. (21 de Abril de 2020). *Banana Pi*. Recuperado el 13 de Mayo de 2020, de <https://bit.ly/3p2gfQi>
- Pi, R. (2 de Abril de 2014). *Raspberry Pi*. Recuperado el 31 de Marzo de 2020, de <https://bit.ly/3aFvMwQ>
- Pi, R. (s.f.). *Raspberry Pi*. Recuperado el 08 de Abril de 2020, de <https://bit.ly/3v81ht8>
- PROGRAM, D. I. (Septiembre de 1981). *tools.ietf.org*. Recuperado el 11 de Enero de 2021, de <https://tools.ietf.org/html/rfc792>
- Project, T. P. (10 de Abril de 2013). *The Parrot Porject*. Recuperado el 08 de Abril de 2020, de <https://bit.ly/3BER0H7>
- Protocolo, R. . (s.f.). *Real Academia Española - Protocolo*. Recuperado el 5 de Mayo de 2020, de <https://bit.ly/3aAsL0V>
- Ptolomeo. (s.f.). Recuperado el 16 de Abril de 2020, de <https://bit.ly/3p4DRnl>
- RAE. (2019). *Real Acedemia Española*. Recuperado el 28 de Abril de 2020, de <https://bit.ly/3oWPlcF>
- Raymond, E. S. (2001). *CATB*. Recuperado el 28 de Abril de 2020, de <https://bit.ly/3ICjBHt>
- RedHat. (s.f.). *RedHat*. Recuperado el 22 de Abirl de 2020, de <https://red.ht/3FUUA6qK>
- Resilient, S. b. (2018). *IBM Company*. Recuperado el 07 de Abril de 2020, de <https://ibm.co/3mPXgWu>

- robot, M. e. (5 de Noviembre de 2015). Recuperado el 5 de Mayo de 2020, de <https://bit.ly/3v8pors>
- ROJAS, E. (26 de Febrero de 2020). *COINTELEGRAPH*. Recuperado el 4 de Mayo de 2020, de <https://bit.ly/3FV1dIB>
- S2Grupo. (10 de 2017). *Security At Work*. Recuperado el 30 de Abril de 2020, de <https://bit.ly/2XaBSSV>
- Said, O., & Mehedi Masud. (2013). Recuperado el 19 de Mayo de 2020, de <https://bit.ly/2XfD3AI>
- Sain, G. (Mayo de 2018). Recuperado el 08 de Abril de 2020, de <https://bit.ly/3v6QZte>
- Salazar, J., & Silvestre, S. (2016). *UPCommons*. Recuperado el 1 de Abril de 2020, de <https://bit.ly/3FKK5ic>
- Salud, R. d. (Junio de 2018). *Revista de la Sociedad Española de Informática y Salud*. Recuperado el 19 de Mayo de 2020, de <https://bit.ly/30fWOJm>
- Sánchez Martelo, C. A. (18 de Agosto de 2015). *ScIELO*. Recuperado el 19 de Mayo de 2020, de <https://bit.ly/3mUKPbl>
- Santos, H. R. (01 de Agosto de 2019). *Incibe Cert*. Recuperado el 11 de Enero de 2021, de <https://bit.ly/3v7TbR2>
- Schiavo, U. F. (3 de Enero de 2014). *Eleven Paths*. Recuperado el 22 de Abril de 2020, de <https://bit.ly/2XfDIHO>
- Security, D. O. (s.f.). *Departament Of Homeland Security*. Recuperado el 08 de Abril de 2020, de <https://bit.ly/3FJCYq6>
- Security, N. (s.f.). *Norton Security*. Recuperado el 28 de Abril de 2020, de <https://nr.tn/3BHjKPJ>
- Security, O. (25 de Noviembre de 2019). *Offensive Security*. Recuperado el 08 de Abril de 2020, de <https://bit.ly/3mTOmXU>
- Security, P. (27 de Septiembre de 2018). *Panda Security*. Recuperado el 22 de Abril de 2020, de <https://bit.ly/2YKpZ7f>
- Security, P. (s.f.). *Panda Security*. Recuperado el 08 de Abril de 2020, de <https://bit.ly/3ay8JEi>
- Semle, A. (s.f.). *Editoriales SRL*. Recuperado el 5 de Mayo de 2020, de <https://bit.ly/3aAv5Ff>
- SHODAN. (2009). *SHODAN*. Recuperado el 2 de Junio de 2021, de <https://bit.ly/3Az2s5W>
- Software, M. p. (2001). *Manifiesto por el Desarrollo Ágil de Software*. Recuperado el 21 de Mayo de 2020, de <https://bit.ly/3DvGuCK>
- Software, O. A. (2020). *Open Automation Software*. Recuperado el 5 de Mayo de 2020, de <https://bit.ly/2YVT4wh>
- SolidRun. (2020). *SolidRun*. Recuperado el 13 de Mayo de 2020, de <https://bit.ly/3FGPbvQ>
- Steven J. Johnston, P. J. (30 de Junio de 2018). *University of Cambridge*. Recuperado el 7 de Mayo de 2020, de <https://bit.ly/30fwXBg>

- system, a. I. (22 de Agosto de 2018). *adafruit learning system*. Recuperado el 2020 de Abril de 1, de <https://bit.ly/3FleOMT>
- Systems, C. (Abril de 2011). CISCO. Recuperado el 19 de Mayo de 2020, de <https://bit.ly/3aAvcRb>
- Systems, C. (2012). *Cisco CCNA Security 1.0*. Systems, Cisco.
- Systems, C. (25 de Marzo de 2015). Recuperado el 28 de Junio de 2021, de <https://bit.ly/3BKwgxO>
- Systems, C. (s.f.). CCNA Routing ans Switching. En *Capítulo 8: Direccionamiento IP*. Systems, Cisco.
- Techopedia. (s.f.). *Techopedia*. Recuperado el 13 de Mayo de 2020, de <https://bit.ly/3AEuVr3>
- TI, E. (14 de Octubre de 2019). Recuperado el 16 de Abril de 2020, de <https://bit.ly/2XbhckH>
- Trigas Gallego, M., & Domingo Troncho, A. (s.f.). *Open Access UOC*. Recuperado el 20 de Mayo de 2020, de <https://bit.ly/3pfcs2n>
- Urueña, A. (2012). *ontsi*. Recuperado el 30 de Junio de 2021, de <https://bit.ly/3v8MOx5>
- Valle, R. J. (Febrero de 2007). *Di-mare*. Recuperado el 15 de Julio de 2021, de <https://bit.ly/3mULav0>
- Vasilis Katos, S. R. (2019). ENISA. Recuperado el 28 de Abril de 2020, de <https://bit.ly/2XdtYZ6>
- Vertiv. (23 de Octubre de 2018). Recuperado el 21 de Abril de 2020, de <https://bit.ly/3AQqg5z>
- Víctor Manuel CastellanosBernal, D. C. (Diciembre de 2019). UNIANDES. Recuperado el 27 de Abril de 2020, de <https://bit.ly/3ayolaL>
- Voices, H. (s.f.). *HPE Voices*. Recuperado el 18 de Mayo de 2020, de <https://bit.ly/3DHtVEf>
- Wang, Y. (2016). *Research Online Publishing*. Recuperado el 28 de Junio de 2021, de <https://bit.ly/3vnovf3>
- Watson, W. T. (s.f.). *Willis Towers Watson*. Recuperado el 19 de Mayo de 2020, de <https://bit.ly/3j2lmw1>
- Winder, D. (19 de Febrero de 2020). *Forbes*. Recuperado el 27 de Abril de 2020, de <https://bit.ly/3azx902>
- ZERODIUM. (s.f.). Recuperado el 21 de Abril de 2020, de <https://bit.ly/3j2lg7D>
- Z-Wave. (s.f.). *Z-Wave*. Recuperado el 7 de Mayo de 2020, de <https://bit.ly/2YKWcv9>

7. ANEXOS

Anexo 1 - Código Fuente

Anexo 2 - Manual del Usuario

Anexo 3 - Manual Técnico

Anexo 4 - Clasificación de Vulnerabilidades IoT

Anexo 5 - Criterios de Aceptación HU

Anexo 6 - Planning Sprint

Anexo 7 - Product Backlog

Anexo 8 - Sprint Backlog

Anexo 9 - Certificado de Registro de Software - Escaneo de Dispositivos

Anexo 10 - Certificado de Registro de Software - Detección de Vulnerabilidades