UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE TELECOMUNICACIONES

TRABAJO DE INTEGRACIÓN CURRICULAR

TEMA:

"PROTOTIPO DE CONTROL PARENTAL BASADO EN UNA PLATAFORMA SBC PARA FILTRADO DE CONTENIDOS MALICIOSOS EN REDES DOMÉSTICAS, BASADO EN LISTAS NEGRAS Y ANÁLISIS HEURÍSTICO."

Trabajo de titulación previo a la obtención del título de Ingeniero en Telecomunicaciones

Línea de investigación: Desarrollo, aplicación de software y cibersecurity (seguridad cibernética)

AUTOR:

Bryan Andrés Arotingo Guandinango

DIRECTOR:

Ing. Fabián Geovanny Cuzme Rodríguez, Msc

Ibarra – Ecuador 2026

UNIVERSIDAD TÉCNICA DEL NORTE
DIRECCIÓN DE BIBLIOTECA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO	
APELLIDOS Y NOMBRES:	AROTINGO GUANDINANGO BRYAN ANDRÉS

DATOS DE LA OBRA	
TÍTULO:	"PROTOTIPO DE CONTROL PARENTAL BASADO EN UNA PLATAFORMA SBC PARA FILTRADO DE CONTENIDOS MALICIOSOS EN REDES DOMÉSTICAS, BASADO EN LISTAS NEGRAS Y ANÁLISIS HEURÍSTICO."
AUTOR (ES):	AROTINGO GUANDINANGO BRYAN ANDRES
FECHA: DD/MM/AAAA	30/07/2026
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	☐ **PREGRADO** ☐ **POSGRADO**
TITULO POR EL QUE OPTA:	INGENIERO EN TELECOMUNICACIONES
DIRECTOR /ASESOR:	ING. FABIÁN GEOVÁNNY CUZME RODRÍGUEZ, MSC ING. CARLOS ALBERTO VÁSQUEZ AYALA, MSC

2. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 30 días del mes de julio de 2026.

EL AUTOR:

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
BRYAN ANDRES AROTINGO GUANDINANGO

.....

Bryan Andrés Arotingo Guandinango

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 30 de Julio de 2026

ING. FABIÁN GEOVANNY CUZME RODRÍGUEZ, MSC
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
FABIAN GEOVANNY CUZME RODRIGUEZ

……

Ing. Fabián Geovanny Cuzme Rodríguez, Msc
C.C.: 1311527012

DEDICATORIA

A Dios, por ser mi guía, darme la fortaleza, sabiduría y salud necesarias para superar cada obstáculo en este camino y alcanzar este logro.

A mi familia, por ser el pilar fundamental en mi vida, por su apoyo incondicional, sacrificio y fe constante en mis proyectos. Su amor y ejemplo han sido el motor que me impulsó a superar cada reto académico y personal.

A mi compañera incondicional, por estar a mi lado en cada etapa, por su comprensión, paciencia y apoyo constante en los momentos más exigentes de este proceso.

A mis verdaderos amigos y compañeros de camino, por su constante apoyo, palabras de aliento y por compartir las largas jornadas de trabajo que hoy se traducen en este logro.

Bryan Andrés Arotingo Guandinango

AGRADECIMIENTO

A la Universidad y a la Carrera de Ingeniería en Telecomunicaciones, por haberme abierto sus puertas y brindarme los conocimientos y herramientas necesarias para mi formación profesional.

Expreso mi sincero reconocimiento a mi Director de Tesis, el Ing. Fabián Cuzme, y a mi Asesor, el Ing. Carlos Vásquez, por su valiosa guía, paciencia, criterio técnico y acompañamiento constante durante el desarrollo de esta investigación.

De igual manera, extiendo mi agradecimiento al Coordinador de Carrera, el Ing. Jaime Michilena, por su gestión, liderazgo y permanente respaldo a los estudiantes en la consecución de nuestras metas académicas.

Finalmente, agradezco a todas las personas que, de manera directa o indirecta, aportaron con su conocimiento y apoyo para la culminación con éxito de este proyecto.

Bryan Andrés Arotingo Guandinango

RESUMEN

El crecimiento del uso de Internet y de los dispositivos móviles ha facilitado el acceso de niños y adolescentes a una gran cantidad de información y servicios digitales. Sin embargo, este acceso también aumenta el riesgo de exposición a contenido inapropiado, sitios maliciosos y otras amenazas que pueden afectar su seguridad. A pesar de la existencia de varias alternativas de control parental, muchas tarjetas se nutren de las suscripciones, las limitaciones para personalizarlas o bien se nutren de servicios externos. Como solución a este problema, la propuesta de esta investigación es el desarrollo de un prototipo de control parental desde una plataforma de Raspberry Pi 5 para supervisar y controlar la navegación en las redes de hogar. El sistema se fundamenta en el uso del filtrado de dominios a través de listas negras y del análisis heurístico de los comportamientos de navegación para verificar patrones de conducta de riesgo, realizar restricción a potencias no permitidos al usuario y generar alertas para el administrador/a de la red.

Se llevó a cabo la implementación combinando Pi-hole, Suricata, FastAPI y MariaDB, además de dos aplicaciones móviles Android para la administración y el monitoreo de la funcionalidad del sistema. Finalmente, las pruebas de funcionamiento lograron comprobar la comunicación entre los diferentes componentes, la aplicación de las políticas de control y la generación de reportes y notificaciones, evidenciando que la opción del prototipo puede ser considerada como una vía de baja invertibilidad para llevar a cabo la supervisión parental en el hogar.

Palabras Clave: Raspberry Pi, API, Análisis heurístico, IA generativa, Filtrado de dominios, Supervisión de tráfico, Pi-hole, iptables, Android.

ABSTRACT

The increasing use of the Internet and mobile devices has enabled children and adolescents to access a vast range of information and digital services. However, such access also increases the risk of exposure to inappropriate content, malicious websites, and other threats that may compromise their online safety. Although several parental control solutions are currently available, many of them depend on paid subscriptions, offer limited customization options, or rely on external services. To address this issue, this research proposes the development of a parental control prototype based on a Raspberry Pi 5 platform to monitor and manage Internet browsing within home networks.

The system is based on domain filtering through blacklists and heuristic analysis of browsing behavior to identify potentially risky patterns, restrict access to unauthorized content, and generate alerts for the network administrator. The prototype was implemented by integrating Pi-hole, Suricata, FastAPI, and MariaDB, along with two Android mobile applications designed for system administration and functionality monitoring.

Finally, functional tests confirmed successful communication among the different system components, the proper enforcement of parental control policies, and the generation of reports and notifications. These results demonstrate that the proposed prototype represents a low-cost and customizable alternative for implementing parental supervision in home network environments.

Keywords: Raspberry Pi, API, Heuristic analysis, Generative AI, Domain filtering, Traffic supervision, Pi-hole, iptables, Android.

LISTA DE CONTENIDOS

Capítulo I: Antecedentes 20

1.1 Tema 20

1.2 Problema 20

1.3 Objetivos 22

1.3.1 Objetivo General 22

1.3.2 Objetivos Específicos 22

1.4 Alcance 23

1.5 Justificación 25

2 Capítulo II: Fundamentación Teórica 27

2.1 Comunicaciones en Redes Domésticas 27

2.1.1 Medios Guiados 27

2.1.2 Comunicaciones Inalámbricas 29

2.2 Seguridad en redes domésticas 31

2.2.1 Cortafuegos o Firewall 31

2.2.2 Firewall de siguiente generación 32

2.2.3 Proxy 33

2.2.4 Control parental 33

1.1.1.8. Control parental a nivel de hardware 34

1.1.1.9. Control parental a nivel de software 34

2.3 Single Board Computer (SBC) 35

2.3.1 Arquitectura SBC 35

2.3.2 Comparación de SBC y SBM 36

2.4 Protocolos de Red 37

2.4.1 DHCP 37

2.4.2 DNS 38

2.5 Protocolos de comunicación y gestión 38

2.5.1 API (Application Programming Interface) 38

2.5.2 API REST sobre HTTPS 38

2.5.3 SSH 39

2.5.4 VPN 39

2.6 Gestión de contenido 40

2.6.1 Software de filtrado y monitoreo en redes domésticas 40

2.6.2 Análisis heurístico 42

3 Capítulo III: Diseño e Implementación 44

3.1 Metodología 44

3.2 Análisis de Requerimientos 45

3.2.1 Situación Actual 45

3.2.2 Estudio de la Población Objetivo 46

3.2.3 Propósito del sistema 47

3.2.4 Trabajos relacionados 48

3.3 Requerimientos 50

3.3.1 Stakeholders 50

3.3.2 Nomenclatura de requerimientos 51

3.3.3 Requerimientos de stakeholders 53

3.3.4 Requerimientos del sistema 54

3.3.5 Requerimientos de la arquitectura 57

3.4 Selección de Hardware y software 60

3.4.1 Hardware 60

3.4.2 Software 63

3.4.3 Software Seleccionado para el Sistema 68

3.5 Diseño del Sistema 70

3.5.1 Arquitectura del Sistema 70

3.5.2 Diagrama de Conexión del Sistema 72

3.5.3 Módulo de Red 74

3.5.4 Módulo de Gestión de Servicios 77

3.5.5 Módulo de Conexión Lógica 79

3.5.6 Módulo de monitorea e interfaz de usuarios 83

3.5.7 Módulo de análisis heurístico 88

3.6 Implementación 93

3.6.1 Implementación de Software 93

3.6.2 Módulo de red 93

3.6.3 Módulo de gestión de servicios 96

3.6.4 Módulo de conexión lógica 104

3.6.5 Módulo de Monitoreo e Interfaz de Usuario 115

3.6.6 Módulo de Análisis Heurístico 130

3.6.7 Implementación de Hardware 136

4 Capítulo IV: Pruebas de Funcionamiento 140

4.1 Plan de Pruebas 140

4.1.1 PS-CL01: Pruebas de Conexión Lógica 142

4.1.2 PS-R01: Prueba de Hostpod 145

4.1.3 PS-R02: Prueba de autenticación de usuario 147

4.1.4 PS-G01: Prueba de bloqueo/desbloqueo de sitios web 149

4.1.5 PS-G02: Prueba de corte y horarios de internet 155

4.1.6 PS-MI01: Prueba de Gestión de Dispositivos 159

4.1.7 PS-MI02: Prueba de alertas y notificaciones 163

4.1.8 PS-AH01: Pruebas del análisis heurístico 169

4.1.9 PH-IR01: Cobertura de la señal 173

4.1.10 PH-RO01: Rendimiento del SBC 178

4.2 Cumplimiento de las pruebas 186

4.2.1 Cumplimiento de requerimientos Stakerholders 186

4.2.2 Cumplimiento de requerimientos del sistema 188

4.2.3 Cumplimiento de requerimiento de la arquitectura del sistema 189

Conclusiones 191

5 Bibliografía 194

LISTA DE TABLAS

Tabla 1 *Técnicas de análisis heurístico* 42

Tabla 2 *Actores involucrados* 51

Tabla 3 *Nomenclatura de los requerimientos* 52

Tabla 4 *Prioridades de los requerimientos del prototipo de control parental* 52

Tabla 5 *Requerimientos de stakeholders* 53

Tabla 6 *Tabla de Requerimientos del sistema* 55

Tabla 7 *Requerimientos de la Arquitectura* 57

Tabla 8 *Requerimientos de SBC* 61

Tabla 9 *Hardware Seleccionado* 62

Tabla 10 *Requerimientos para el DNS* 64

Tabla 11 *Requerimientos de Bases de datos* 65

Tabla 12 *Requerimientos de API* 66

Tabla 13 *Requerimientos para la aplicación móvil* 67

Tabla 14 *Software seleccionado* 69

Tabla 15 *Estructura y distribución de endpoints* 82

Tabla 16 *Comparativa de versiones de Pi hole* 100

Tabla 17 *Tecnologías de la aplicación de cliente ChildGuard* 123

Tabla 18 *Especificación de interfaces y puertos* 139

Tabla 19 *Nomenclatura del plan de pruebas* 141

Tabla 20 *Resumen de plan de pruebas* 141

Tabla 21 *Conectividad y disponibilidad de la API REST* 142

Tabla 22 *Capacidad de monitoreo de dispositivos* 145

Tabla 23 *Pruebas de autenticación* 147

Tabla 24 *Bloqueo y desbloqueo web* 149

Tabla 25 *Corte de Internet* 155

Tabla 26 *Monitoreo ChildGuard* 159

Tabla 27 *Prueba de alertas y notificaciones* 163

Tabla 28 *Análisis heurístico y reporte general* 169

Tabla 29 *Prueba de Cobertura de la señal* 173

Tabla 30 *Sitos de pruebas de la casa para medir la intensidad de la señal* 178

Tabla 31 *Rendimiento del SBC mediante pruebas de respuesta* 178

Tabla 32 *Resultados durante la prueba de rendimiento* 180

Tabla 33 *Tabla comparativa con ambos escenarios de carga* 182

Tabla 34 *Prueba de rendimiento del SBC bajo carga* 183

Tabla 35 *Comparativa de eventos antes y durante la carga* 185

Tabla 36 *Cumplimiento de Requerimientos de Stakeholders* 187

Tabla 37 *Cumplimiento de requerimientos del sistema* 188

Tabla 38 *Cumplimiento de los requerimientos de la arquitectura* 190

Tabla 39 : Ficha Bibliográfica de estudios con referencia a control parental y rede domesticas 199

LISTA DE FIGURAS

Figura 1 *Arquitectura de la red* 24

Figura 2 Firewall tradicional 32

Figura 3 *Servidor Proxy* 33

Figura 4 *Arquitectura típica de una computadora de placa única.* 36

Figura 5 *Arquitectura del control parental* 71

Figura 6 *Diagrama de conexión del proyecto* 74

Figura 7 *Diagrama de bloques del módulo de red* 74

Figura 8 *Diagrama de flujo del módulo de Red (conexión y autenticación)* 76

Figura 9 *Diagrama de flujo de módulo de Gestión de Servicios* 78

Figura 10 *Interacción de API REST* 79

Figura 11 *Diagrama general de flujo de conexión lógica (API)* 81

Figura 12 *Diagrama de flujo del módulo de la interfaz del adminstrador* 84

Figura 13 *Estructura de la aplicación* 85

Figura 14 *Diagrama de la aplicación a monitorear (Childguard)* 87

Figura 15 *Estructura de la aplicación Childguard* 88

Figura 16 *Diagrama del análisis heurístico* 90

Figura 17 *Diagrama de bloques sobre el método de análisis heurístico* 91

Figura 18 *Hostpot de Pi 5 revisar el canal* 95

Figura 19 *Configuración de DHCP con dnsmasq* 96

Figura 20 *Configuración del docker compose* 98

Figura 21 *Inicialización del docker* 98

Figura 22 *Reglas de Reenvío (FORWARD)* 101

Figura 23 *Regla NAT: Enmascaramiento de red* 101

Figura 24 *Base de datos MariaDB* 104

Figura 25 *Inicialización de la API* 106

Figura 26 *Endpoint de autenticación* 107

Figura 27 *Conexión de servicios y configuración del sistema* 108

Figura 28 *Recarga de listas de Pi-hole* 109

Figura 29 *Endpoint de lista negra* 110

Figura 30 *Endpoint Bloqueo con IA* 111

Figura 31 *Endpoint de control de dispositivos* 112

Figura 32 *Endpoint de generación de reportes con IA* 113

Figura 33 *Generar reporte como PDF* 114

Figura 34 *Dependencias de la app del administrador* 115

Figura 35 *Navegación y estructura principal de la aplicación* 116

Figura 36 *Direcciones para la conexión local o externa* 117

Figura 37 *Login de la APP* 118

Figura 38 *Screen de dispositivos conectados* 119

Figura 39 *Screen de dispositivos conectados* 120

Figura 40 *Sistema de notificaciones en tiempo real* 121

Figura 41 *Sección de alertas de la app* 122

Figura 42 *Sección de la vinculación con la aplicación del administrador* 125

Figura 43 *Sección del servicio en segundo plano de Childguard* 126

Figura 44 *Sección de bloqueo de aplicaciones mediante AccessibilityService* 127

Figura 45 *Sección del reporte de actividad al servidor* 128

Figura 46 *Sección del inicio automático al encender el celular* 129

Figura 47 *Interfaz de la aplicación de Childguard* 129

Figura 48 *Sección de detección de patrones* 130

Figura 49 *Sección de lectura del registro de Pi hole* 131

Figura 50 *Sección de identificación de aplicaciones con VPN* 131

Figura 51 *Horarios de navegación* 132

Figura 52 *Sección del análisis para el puntaje de los dispositivos* 133

Figura 53 *Sección de reportes y análisis heurístico* 134

Figura 54 *Reporte del Análisis Heurístico* 136

Figura 55 *Conexión posterior de los equipos de red* 137

Figura 56 *Servicio de API activo* 143

Figura 57 *Verificación de API* 144

Figura 58 *Dispositivos activos en la red* 144

Figura 59 *Asignación de dirección IP* 145

Figura 60 *Pantalla de dispositivos conectados* 146

Figura 61 *Prueba de conectividad* 147

Figura 62 *Verificación del administrador* 148

Figura 63 *Bloqueo de la página web mediante la aplicación* 150

Figura 64 *Página web bloqueado y agregado a la lista negra* 151

Figura 65 *Registro de huellas digitales de Facebook.* 151

Figura 66 *Página de red social bloqueada* 152

Figura 67 *Verificación de resolución DNS* 153

Figura 68 *Desbloqueo de la página web* 153

Figura 69 *Resolución DNS autorizada por Pi-hole* 154

Figura 70 *Horarios de internet* 156

Figura 71 *Dispositivo bloqueado por horarios* 156

Figura 72 *Cortar el internet mediante app* 157

Figura 73 *Corte de internet a cliente* 157

Figura 74 *Creación de regla para denegar acceso a internet* 158

Figura 75 *Restauración del internet* 158

Figura 76 *Código de vinculación hacia el control parental* 160

Figura 77 *Dispositivo vinculado correctamente* 161

Figura 78 *Bloqueo de aplicación desde la app de administrador* 161

Figura 79 *Verificación dentro del reporte general* 162

Figura 80 *Intento de ingreso a página de lista negra y prohibida* 164

Figura 81 *Notificación de intento a sitios prohibidos* 164

Figura 82 *Bloqueo de aplicaciones* 165

Figura 83 *Notificación de app bloqueada* 166

Figura 84 *Prueba de ataque a puertos* 167

Figura 85 *Notificación de ataque a puertos* 167

Figura 86 Alerta de ataque de puertos 168

Figura 87 *Consulta al Score del análisis heurístico* 170

Figura 88 *Generación de reporte mediante la Aplicación de administrador* 171

Figura 89 *Generación de PDF por consola* 172

Figura 90 *PDF con los registro y reportes del día* 172

Figura 91 *Uso de Wifianalyzer dentro de la red domestica* 174

Figura 92 *Intensidad de la señal a 1 metro de distancia* 175

Figura 93 *Intensidad mediada a 5m de distancia* 175

Figura 94 *Intensidad mediada a 8m de distancia* 176

Figura 95 *Intensidad mediada a 10m de distancia* 177

Figura 96 *Intensidad mediada a 13m de distancia* 177

Figura 97 *Prueba de rendimiento del SBC dentro de un escenario controlado* 180

Figura 98 *Prueba de rendimiento con 50 clientes simulados* 181

Figura 99 *Verificación del estado actual de la SBC antes de las pruebas* 184

Figura 100 *Prueba de estrés a la SBC* 185

Figura 101 Resultados de la primera pregunta 203

Figura 102 Resultados de la segunda pregunta 204

Figura 103 Resultados de la tercera pregunta 205

Figura 104 Resultados de la Cuarta pregunta 205

Figura 105 Resultados de la quinta pregunta 206

Figura 106 Resultados de la sexta pregunta 206

Figura 107 Resultados de la séptima pregunta 207

Figura 108 Resultados de la octava pregunta 207

Figura 109 Resultados de la novena pregunta 208

Figura 110 Resultados de la décima pregunta 208

Figura 111 Resultados de la 11va pregunta..... 209

Figura 112 Resultados de la 12va pregunta 209

Capítulo I: Antecedentes

En el presente capítulo se aborda la problemática relacionada con el acceso no supervisado de niños y adolescentes a contenidos digitales inapropiados. Se presenta una propuesta de solución tecnológica mediante un prototipo de control parental basado en un SBC, el cual permitirá gestionar y filtrar el contenido en redes domésticas. Asimismo, se detallan los objetivos, el alcance del proyecto y la justificación que sustenta su importancia en el contexto actual.

1.1 Tema

Prototipo de control parental basado en una plataforma SBC para filtrado de contenidos maliciosos en redes domésticas, basado en listas negras y análisis heurístico.

1.2 Problema

El internet ha cambiado radicalmente la manera en que las personas indagan fuentes de información o simplemente acceden a ellas, ya sea para aprender, comunicarse, investigar o profundizar en determinados temas [1]. Sin embargo, en la mayoría de los casos, el principal objetivo es el entretenimiento en distintas áreas. Durante la pandemia de COVID-19, el uso de internet y de dispositivos, tanto móviles como ordenadores, tuvo un notable incremento en Ecuador.

Según el INEC, el porcentaje de personas que utilizaban internet aumentó en 11,5 puntos porcentuales a nivel nacional en 2020, en comparación con 2019. Este crecimiento fue impulsado principalmente por la implementación del teletrabajo y la educación virtual durante los periodos de confinamiento[2].

Sumado a ello, el uso de teléfonos inteligentes o smartphones también se incrementó, especialmente entre los niños. El INEC dio a conocer que, durante la pandemia, el uso de

smartphones ascendió del 76,8 % al 81,8 % a nivel nacional, lo que representó un aumento de 5,0 puntos porcentuales. En el caso específico de los niños, el incremento fue de 4,4 puntos porcentuales en comparación con 2019 [3].

A raíz de la pandemia, la conectividad aumentó considerablemente, lo que, sumado al incremento en el uso de dispositivos móviles, permitió dar continuidad a actividades tanto laborales como educativas. No obstante, también se expuso la integridad de las personas, especialmente de los niños y adolescentes. Entre los riesgos se encuentran el acceso a contenidos inapropiados, el ciberacoso, el robo de información, los contactos peligrosos, la publicidad engañosa, entre otros [4]

Aunado a lo anterior, los niños son más vulnerables a publicidad maliciosa. Muchas plataformas digitales incluyen anuncios disfrazados de contenido amigable o juegos, lo que hace más fácil que los niños caigan en esas trampas. Esta exposición puede llevarlos a interactuar con productos inapropiados para su edad, como juegos de azar, servicios para adultos o incluso realizar compras no autorizadas [5].

Dado lo anterior, se evidenció que muchos padres no están capacitados para controlar el contenido al que acceden sus hijos, ni el tiempo que permanecen conectados. Esto es especialmente crítico considerando que los niños, desde niveles iniciales, necesitan una guía directa para navegar en internet de manera segura [4]. A pesar de que actualmente existen soluciones de software para el control parental, muchas son ignoradas, ya que los adultos, especialmente los tutores de niños no cuentan con los conocimientos necesarios para configurarlas adecuadamente [4].

Para dar respuesta a esta problemática, se propone el desarrollo de un prototipo de sistema de control parental para redes domésticas, basado en un SBC (Single Board Computer). Este dispositivo será autónomo, de bajo costo, fácil de instalar y manejar, incluso por usuarios sin

conocimientos técnicos avanzados. Su propósito principal es permitir a los padres monitorear, restringir y gestionar el acceso a contenidos digitales de manera eficiente y personalizada.

Comprender esta situación resulta fundamental para salvaguardar la integridad emocional, psicológica y social de niños y adolescentes, quienes son particularmente vulnerables ante los riesgos del entorno digital. Ignorar estas amenazas puede derivar en consecuencias severas, como la exposición a contenido inapropiado, ciberacoso o pérdida de privacidad. Por ello, implementar una solución tecnológica accesible y efectiva contribuirá significativamente al fortalecimiento de la seguridad digital en los hogares y a una navegación responsable por parte de los menores.

1.3 Objetivos

1.3.1 *Objetivo General*

Implementar un prototipo de control parental basado en SBC para la detección y filtrado de contenidos maliciosos en redes domésticas, utilizando técnicas de listas negras y análisis heurístico.

1.3.2 *Objetivos Específicos*

- Investigar los fundamentos teóricos relacionados con control parental, plataformas SBC, filtrado de contenidos y análisis heurístico, para sustentar técnicamente el desarrollo del prototipo.
- Diseñar la arquitectura de hardware y software del prototipo de control parental, seleccionando los componentes, algoritmos y métodos de comunicación más adecuados.
- Desarrollar el firmware de la SBC, integrando funciones de filtrado de contenido mediante palabras clave, listas negras, bloqueo de sitios, programación de acceso y monitoreo en general.

- Realizar pruebas de funcionamiento, rendimiento y efectividad del prototipo en un entorno de red doméstica, analizando los resultados obtenidos durante estos procesos.

1.4 Alcance

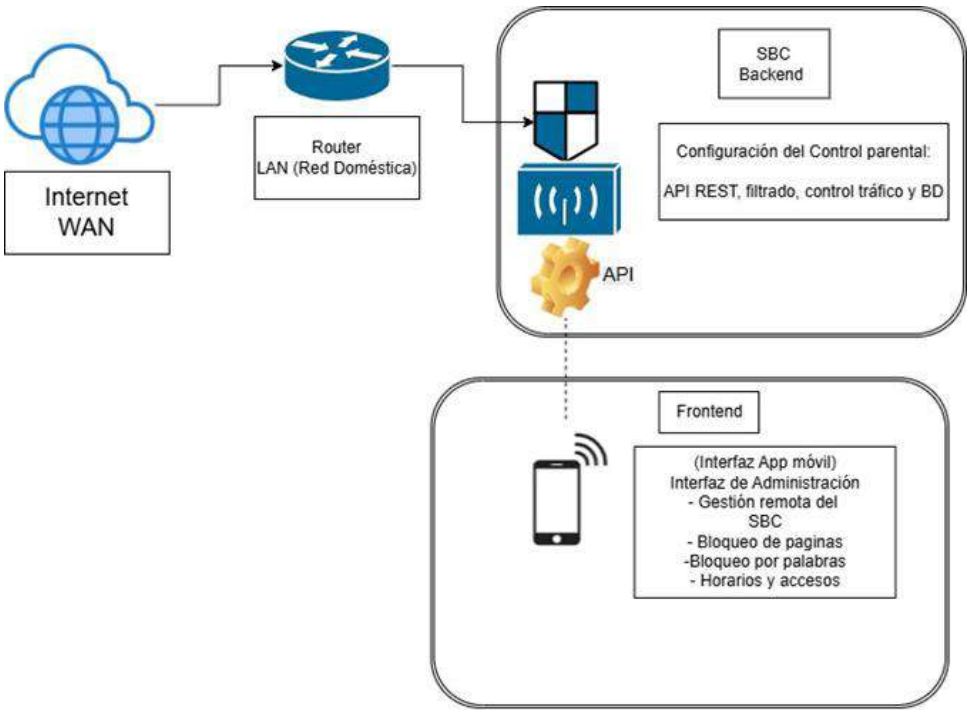
El presente proyecto contempla el diseño e implementación de un prototipo de control parental para el establecimiento de límites dentro de una red doméstica. El trabajo se fundamentará en el modelo en cascada, el cual se describe como un enfoque que divide los procesos de desarrollo en fases secuenciales, cada una de ellas debe finalizarse antes de iniciar la siguiente, lo que garantiza un desarrollo ordenado [6].

Como primera fase, se recolectará y analizará información sobre los conceptos fundamentales relacionados con el control parental, por ejemplo, los riesgos comunes a los que están expuestos los menores al navegar por internet, con el fin de plantear adecuadamente las funciones necesarias para un sistema de control parental. Posteriormente, se analizarán las diferentes herramientas disponibles en el mercado, lo cual permitirá evaluar las opciones existentes y determinar si sus características son óptimas para el desarrollo del prototipo. Esta fase permitirá establecer las bases teóricas y técnicas que sustentarán el diseño del prototipo.

Siguiendo con la siguiente fase se procederá al diseño de la arquitectura del prototipo, abarcando tanto el hardware como el software. En esta fase se seleccionarán los componentes electrónicos, la plataforma SBC más adecuada, los algoritmos de filtrado y bloqueo, así como los métodos de comunicación y la estructura de la interfaz de administración. Además, en esta fase se define la estructura visual de la arquitectura del frontend y backend. Esta disposición permite la comunicación adecuada entre el SBC y la aplicación que utiliza el usuario final [5].

Figura 1

Arquitectura de la red



Nota: La figura muestra el esquema de conexión de los componentes que integran el prototipo de control parental.

Después de ello, se desarrollará el firmware necesario para la plataforma SBC, integrando funciones como el filtrado por palabras clave, bloqueo mediante listas negras, bloqueo de anuncios, restricciones por horario y otras funciones básicas de control parental. También se desarrollará un interfaz de usuario sencilla, donde se realice la gestión de control de la red, para que los padres o tutores sin conocimientos técnicos puedan usarlo sin problema.

Por último, se realizan pruebas de funcionalidad dentro de un entorno de red doméstico, evaluando aspectos como el rendimiento, la efectividad del filtrado y la facilidad de uso para personas sin formación técnica. Estas pruebas incluirán una comparación entre la navegación con

y sin el sistema activo, como parte de un enfoque cuasi-experimental que permitirá validar el prototipo.

Este proyecto se desarrollará con fines académicos y como base para futuras investigaciones o ampliaciones que permitan su escalabilidad o comercialización. La solución final busca demostrar la viabilidad técnica de un sistema de control parental autónomo, eficiente y accesible, orientado a hogares que desean mejorar la protección digital de sus miembros más jóvenes.

1.5 Justificación

Actualmente, el acceso a internet es una necesidad importante que impacta el entorno educativo, laboral y social. No obstante, dicho acceso conlleva riesgos significativos, especialmente para niños y adolescentes que, al navegar libremente, pueden exponerse a contenidos inadecuados, prácticas delictivas cibernéticas, pérdida de privacidad y vulnerabilidades emocionales [7]. La situación se agravó durante la pandemia de COVID-19, cuando el uso de dispositivos conectados a la red aumentó exponencialmente en Ecuador, como lo evidencian los datos proporcionados por el Instituto Nacional de Estadísticas y Censos [3]. Este fenómeno puso en evidencia la falta de preparación de muchos padres para gestionar de forma segura la conectividad de sus hijos.

En referencia con estas preocupaciones, la Unión Internacional de Telecomunicaciones (UIT) ha publicado guías para la protección infantil en internet, donde se establece la necesidad de que los entornos digitales ofrezcan opciones de seguridad en el entorno familiar que sea accesibles, seguras y adaptables al contexto de cada país [8]. Estas directrices promueven el desarrollo de herramientas tecnológicas que ayuden a tutores y padres a monitorear la actividad en línea de sus hijos, especialmente en redes domésticas.

Según la [9], es fundamental establecer estrategias y herramientas que impulsen un uso seguro y responsable de internet desde la infancia, garantizando no solo el acceso a la información, sino también la protección ante los peligros del entorno digital. Por su parte, la [10], señala que resguardar a los niños en el ámbito virtual exige una participación conjunta entre los gobiernos, las familias y el desarrollo tecnológico. En este contexto, implementar soluciones asequibles al entendimiento de cualquier individuo contribuye especialmente cuando los adultos presentan limitaciones en conocimientos digitales.

En este sentido [11], destacan el creciente interés académico en el control parental, subrayando su importancia frente a los riesgos digitales infantiles y la necesidad de soluciones innovadoras, accesibles y efectivas para proteger a los menores. Por otro lado, destaca que las herramientas de control parental deben ser seguras, personalizables, fáciles de usar, proteger la privacidad infantil, emitir alertas inteligentes y fomentar la participación del menor con transparencia y confianza [12].

Además, estudios como el de [13],revela que muchos sistemas de control parental presentan fallas críticas de seguridad, exponiendo datos sensibles de niños y padres, lo que evidencia la necesidad de soluciones más seguras. Por ello se requiero una solución segura, con cifrado y que los datos no sean vulnerables.

Esta propuesta está alineada con los principios de protección de la niñez, promoción del uso responsable de las tecnologías, y el fomento de entornos digitales seguros. Además, contribuye al cierre de brechas tecnológicas en el ámbito familiar, al ofrecer una herramienta que no requiere conocimientos técnicos avanzados por parte del usuario final [14]. Su implementación no solo responde a una necesidad social apremiante, sino que también aporta valor al campo de la ingeniería electrónica y la ciberseguridad doméstica.

Capítulo II: Fundamentación Teórica

Este capítulo presenta los conceptos fundamentales que respaldan el desarrollo del prototipo propuesto. Se tratan aspectos generales sobre el funcionamiento de las redes que se emplean en el hogar, así como el concepto de control parental y sus posibles formas de aplicación. Se estudia también el uso de plataformas tecnológicas adecuadas para este tipo de soluciones y se repasan algunos mecanismos que permiten la supervisión remota y el refuerzo de la ciberseguridad. Esta base teórica se convierte en la referencia que permite entender e implementar el sistema que se plantea.

2.1 Comunicaciones en Redes Domésticas

En el entorno de una red doméstica, la comunicación entre los diferentes dispositivos, como los equipos inteligentes y los dispositivos IoT, puede establecerse mediante medios guiados como cables Ethernet o medios no guiados como Wi-Fi o Bluetooth. Estas redes se caracterizan por su fácil implementación, costos accesibles, escalabilidad, compatibilidad con múltiples dispositivos y soporte para servicios multimedia y domóticos [15].

2.1.1 Medios Guiados

Dentro del ámbito de la comunicación, al principio, solo se utilizaban medios alámbricos o guiados. Actualmente, existen tecnologías inalámbricas, pero los medios guiados, tales como el cable de par trenzado o la fibra óptica, siguen siendo muy utilizados, dado su alto grado de fiabilidad. Mediante los medios guiados se puede confinar la señal a trayectorias concretas, lo que conlleva una mayor estabilidad en la transmisión y una predecible: sin duda alguna, estos dos aspectos son imprescindibles en escenarios en donde prima la exactitud y la continuidad del servicio. Ethernet utiliza medios guiados definidos por el estándar IEEE 802.3. [16].

1.1.1.1. IEEE 802.3

Actualmente, Ethernet sigue siendo la base fundamental de las redes cableadas, debido a su fiabilidad, estabilidad y alto rendimiento en entornos interiores. Se trata de un protocolo de capa de enlace estandarizado por el grupo de trabajo IEEE 802.3, el cual permite la transmisión de datos dentro de redes LAN mediante el uso de direcciones físicas MAC. Su diseño y funcionamiento se encuentran regulados por el estándar IEEE 802.3, el cual describe los requisitos técnicos tanto del nivel físico como de la subcapa MAC, definiendo aspectos como la estructura de las tramas, los esquemas de señalización, las tasas de transferencia y el método de acceso al medio basado en CSMA/CD. A lo largo de su desarrollo, esta tecnología ha incorporado mejoras en el tipo de cableado y en las técnicas de modulación, permitiendo alcanzar velocidades que van desde el gigabit hasta capacidades superiores de cientos de gigabits por segundo [17].

1.1.1.2. IEEE 802.3i

El estándar IEEE 802.3i define la implementación denominada 10BASE-T, la cual posibilita la comunicación a una tasa de 10 Mbps empleando cableado de par trenzado en una estructura de red con forma de estrella. Esta especificación representó un cambio relevante respecto al uso de cable coaxial, al ofrecer una alternativa más económica, flexible y sencilla de desplegar [15].

1.1.1.3. IEEE 802.3 u

Identificado como Fast Ethernet, amplía la velocidad de transferencia hasta los 100 Mbps sin perder compatibilidad con implementaciones previas. Esta versión incorpora alternativas físicas como 100BASE-TX, 100BASE-FX y 100BASE-T4, respondiendo a la creciente necesidad de mayor ancho de banda generada por contenido multimedia y aplicaciones colaborativas, manteniendo el mismo formato de trama y el método CSMA/CD[15].

1.1.1.4. IEEE 802.3 ab

Según Stalling [15],el estándar IEEE 802.3ab establece la norma 1000BASE-T, permitiendo alcanzar velocidades de 1 Gbps mediante cableado de cobre Categoría 5e o superior, lo que representa una ventaja significativa al aprovechar infraestructura existente. Esta especificación se consolidó como la más adoptada en redes domésticas y corporativas debido a su alta relación costo rendimiento y su compatibilidad con equipos actuales como switches, servidores y estaciones de trabajo.

2.1.2 Comunicaciones Inalámbricas

La comunicación inalámbrica ha revolucionado significativamente el mundo de las telecomunicaciones, convirtiéndose en una de las formas de conexión más utilizadas a nivel global. Esta evolución responde a la creciente necesidad de mantener dispositivos interconectados sin la dependencia de cables, lo cual ha impulsado el desarrollo de tecnologías más flexibles y accesibles. En este contexto, las redes de área local inalámbricas han ganado gran protagonismo, siendo ampliamente implementadas en diversos entornos como hogares, instituciones educativas escuelas y universidades, oficinas, así como en espacios públicos como plazas y centros comerciales [15].

1.1.1.5. IEEE 802.11

Para optimizar la comunicación de datos sin la necesidad de modificar cableado físico, se han desarrollado diversas técnicas que facilitan la conectividad. Tras el auge de Ethernet, surgió el estándar IEEE 802.11, conocido comercialmente como WiFi. Este trabaja en las capas de enlace de datos y física, permitiendo la implementación de redes inalámbricas [17].

En las redes inalámbricas definidas por el estándar IEEE 802.11, se distinguen dos niveles de servicio. Por un lado, se encuentra el Basic Service Set (BSS), que representa una célula conformada por uno o varios dispositivos, generalmente gestionada por un punto de acceso (AP).

Por otro lado, el Extended Service Set (ESS) agrupa múltiples BSS interconectados mediante un sistema de distribución. Para mejorar el ancho de banda, la eficiencia de transmisión y la resistencia a interferencias, estas redes emplean técnicas de modulación como FHSS (Frequency Hopping Spread Spectrum), DSSS (Direct Sequence Spread Spectrum) y OFDM (Orthogonal Frequency Division Multiplexing) [16].

1.1.1.6. IEEE 802.11 a/b/g

Basado en la información presentada por [15], en el proceso de desarrollo de los estándares inalámbricos IEEE 802.11, la variante 802.11b permitió alcanzar tasas de transmisión de hasta 11 Mbps utilizando la frecuencia de 2.4 GHz. Más adelante surgió 802.11g, que continuó trabajando sobre la misma banda, pero elevó el rendimiento hasta 54 Mbps, garantizando interoperabilidad con dispositivos previos.

Finalmente, 802.11n representó un avance significativo al incorporar tecnologías como MIMO, la ampliación del ancho del canal y la posibilidad de operar tanto en 2.4 GHz como 5 GHz, lo que favoreció velocidades superiores, mejor cobertura y un desempeño más estable en redes con múltiples usuarios conectados simultáneamente.

1.1.1.7. IEEE 802.11 ac

Según Stalling [15], el estándar IEEE 802.11ac opera en la banda de 5 GHz y fue creado como una evolución directa de 802.11n, manteniendo compatibilidad, pero aumentando significativamente el rendimiento. Esto se logra mediante tres mejoras principales: incremento del ancho de canal hasta 160 MHz, el uso de una modulación más eficiente basada en 256-QAM con OFDM, y la ampliación del esquema MIMO hasta ocho flujos espaciales. Gracias a estos avances, el estándar proporciona velocidades cercanas al nivel gigabit y mejor aprovechamiento del espectro, aunque con mayor susceptibilidad al ruido y disminución del alcance cuando se utilizan

modulaciones de alta densidad. Además, incorpora MU-MIMO para atender múltiples dispositivos en forma simultánea y requiere el uso de tramas agregadas A-MPDU para optimizar la eficiencia del canal inalámbrico.

2.2 Seguridad en redes domésticas

Actualmente, el uso de Internet es casi universal, todas las personas interactúan con dispositivos conectados a la red, ya sea de manera consciente o no. Sin embargo, son pocos quienes realmente comprenden cómo proteger su red doméstica mediante herramientas como routers configurados con medidas de seguridad, firewalls o proxys. Aunque algunos usuarios son conscientes de que la red puede representar ciertos riesgos, una gran parte de la población hace uso de ella sin tener conocimientos mínimos sobre ciberseguridad [18].

La incorporación de dispositivos IoT incrementa los riesgos, ya que pueden divulgar información sensible a través de ataques directos o técnicas de ingeniería social, como suplantación, phishing o publicidad engañosa [19].

En cuanto a seguridad, la red doméstica puede gestionarse mediante la implementación de firewalls, filtros por DNS, VPN y servidores proxy, los cuales permiten filtrar el tráfico entrante y saliente, así como restringir el acceso a contenido no autorizado. Adicionalmente, es recomendable segmentar la red principal para una mejor optimización de la seguridad [13].

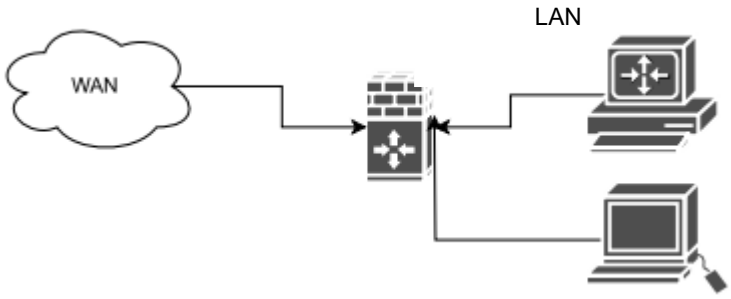
2.2.1 Cortafuegos o Firewall

Según el estudio realizado por [20] , una de las soluciones más eficaces para la seguridad de red es el uso de firewalls. Estos tipos de dispositivos filtran el tráfico que corre por la red, sabiendo qué puede o no recorrerla. La política de la seguridad del cortafuegos se fundamenta en ciertos aspectos como suelen ser las direcciones IP, puertos y protocolos. En comparación con los del siguiente nivel, los de nueva generación permiten una serie de funciones avanzadas que añaden

reglas basadas en el nombre del usuario o la aplicación, en vez de solamente basarse en la programación técnica.

Figura 2

Firewall tradicional



Nota. Representación básica de un firewall en una red LAN. Autoría propia.

2.2.2 Firewall de siguiente generación

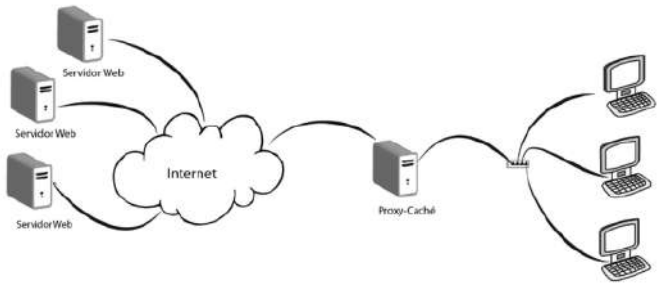
Un firewall de siguiente generación (NGFW) es una plataforma de seguridad diseñada para ofrecer un nivel de protección más avanzado que los cortafuegos tradicionales, ya que no solo filtra el tráfico por puertos, direcciones o protocolos, sino que analiza el contenido, el contexto y el comportamiento del flujo de datos. Este tipo de tecnología integra funciones como inspección profunda de paquetes, identificación de aplicaciones, control basado en identidad, machine learning para detección temprana, prevención automática de intrusiones, análisis de tráfico cifrado y correlación con inteligencia de amenazas en tiempo real. De esta forma, un NGFW permite detectar ataques modernos como malware polimórfico, evasiones, explotación de vulnerabilidades, comunicaciones no autorizadas y uso indebido de aplicaciones, proporcionando visibilidad detallada y capacidad de respuesta proactiva dentro de entornos domésticos, empresariales o críticos [1].

2.2.3 Proxy

En relación con el proxy, este funciona como un intermediario entre el navegador del usuario y el sito al que se desea acceder. El tráfico de red es dirigido primero a través del proxy, lo que permite la inspección, filtrado y monitoreo del flujo de datos antes de alcanzar su destino final. No obstante, la utilización de certificados raíz no confiables o una configuración incorrecta de los mismos puede generar vulnerabilidades críticas en la seguridad. Por esta razón, es imprescindible que la gestión y administración de estas herramientas se realice de manera rigurosa para garantizar la integridad y confidencialidad de las comunicaciones [1].

Figura 3

Servidor Proxy



Nota. Implementación de un servidor proxy, obtenido por Gómez et al. (2012).

2.2.4 Control parental

El acceso de los menores a internet ha aumentado significativamente en los últimos años. Paralelamente, se incrementa la necesidad de implementar estrategias que permitan proteger su bienestar frente a los riesgos del entorno digital. Una de estas estrategias es el control parental, el cual se presenta como una solución eficaz para supervisar el uso de los recursos tecnológicos.

Según [22], el control parental hace referencia a métodos o herramientas empleadas para monitorear o limitar el acceso a internet en dispositivos inteligentes, como teléfonos móviles,

tabletas o computadoras. Además, se concibe como un recurso que fortalece el vínculo entre padres e hijos, al permitir una participación en el uso responsable de la tecnología.

Por su parte, [23] lo describe como un mecanismo clave para salvaguardar la integridad de los menores en su uso cotidiano de la tecnología. Esta herramienta permite analizar y clasificar el contenido digital, generando alertas ante posibles amenazas, lo que contribuye directamente a la protección de la salud mental y emocional de los menores en el entorno digital actual.

1.1.1.8. *Control parental a nivel de hardware*

En un principio, se optó por gestionar la red a través de soluciones basadas en software, debido a que resultaban más prácticas para realizar configuraciones específicas. Con el tiempo se consideró una alternativa más técnica y robusta que consiste en implementar el control parental a nivel de hardware como una solución centralizada. Esta modalidad consiste en implementar el sistema directamente en routers o dispositivos de red, lo que permite configurarlo una sola vez para que administre, filtre y controle todo el tráfico sin requerir software adicional. Al igual que cualquier sistema de control parental, su propósito es supervisar el uso de internet, aplicando restricciones y políticas de acceso dentro del entorno de la red doméstica [24].

1.1.1.9. *Control parental a nivel de software*

El control parental a nivel de software cumple funciones similares a las implementadas mediante hardware, sin embargo, se basa en aplicaciones o programas instalados directamente en dispositivos inteligentes. Una de sus principales ventajas es que no requiere conocimientos técnicos avanzados para su configuración, ya que su uso es comparable al de cualquier aplicación convencional. Actualmente, existen diversas soluciones disponibles en el mercado, como Qustodio o Family Link, que ofrecen recursos eficaces para la supervisión y gestión del uso digital por parte de los menores [25].

2.3 Single Board Computer (SBC)

En los últimos años, el uso de las computadoras de placa única (SBC) se ha vuelto cada vez más común en diversas áreas, como oficinas de entidades financieras y entornos domésticos. Entre sus características más destacadas se encuentran la eficiencia, la portabilidad y la compatibilidad con distintos sistemas operativos y sensores. Estos dispositivos están conformados por una única placa de circuito en la que se integran los componentes principales, tales como salida el procesador, la memoria RAM, la memoria ROM, los sockets y los dispositivos de entrada/salida [26]

Según [27], su diseño optimizado ha sido un factor clave para el crecimiento dentro del ecosistema del Internet de las Cosas (IoT), ya que dispositivos como la Raspberry Pi se emplean ampliamente en ámbitos educativos, en procesos de automatización y en proyectos relacionados con la ciberseguridad.

2.3.1 Arquitectura SBC

Las computadoras de placa única se caracterizan por su arquitectura basada en un System on Chip (SoC), lo cual permite integrar en un solo circuito diversos subsistemas como la CPU, GPU y controladores de red. Esta integración permite que todo el sistema esté optimizado tanto en consumo energético como en espacio, lo que resulta ideal para aplicaciones embebidas[26].

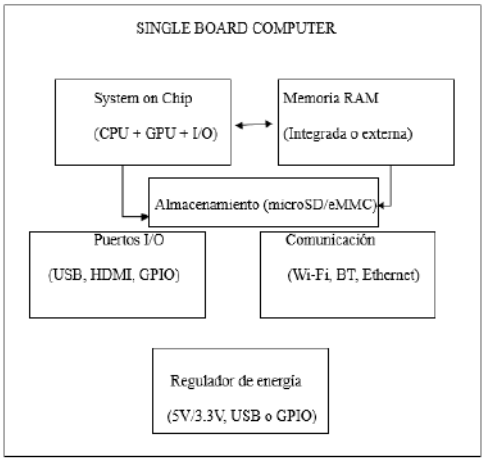
Por otro lado, los CPU convencionales dependen de componentes discretos como el reloj de tiempo real (RTC), mientras que en las SBC estos suelen ser reemplazados por osciladores internos compartidos, que permiten sincronizar diferentes operaciones [28].

A continuación, se presenta una arquitectura común de las computadoras de placa única, en la cual se destacan los elementos principales como el SoC, la memoria RAM, el

almacenamiento, que usualmente se implementa mediante tarjetas microSD o memoria eMMC, los puertos de entrada/salida (E/S) y su sistema de comunicación, ya sea cableado o inalámbrica.

Figura 4

Arquitectura típica de una computadora de placa única.



Nota. Arquitectura básica de una SBC basado en SoC. Elaboración propia.

2.3.2 Comparación de SBC y SBM

Las SBC son una solución versátil, optima y económica para la implementación de sistemas destinados tanto a la adquisición de datos como al monitoreo general. Sin embargo, no todas las SBC poseen los requisitos técnicos necesarios para satisfacer demandas específicas, por lo que es fundamental evaluar y comparar distintas arquitecturas de placas únicas [29].

Además de las SBC, existen las Single Board Microcontrollers (SBM), que conforman otra categoría dentro de las plataformas de desarrollo basadas en placas únicas. Según [27], las Single Board Microcontroller (SBM), se basan en microcontroladores de bajo costo y, a diferencia de las SBC, no incorporan un sistema operativo. Esto se debe a que las SBM están enfocadas principalmente en la interacción directa con el hardware, lo que implica que su programación se

realiza mayormente en lenguaje C o C++ utilizando entornos de desarrollo integrados (IDE) como el IDE Arduino.

La mayoría de las plataformas SBM son open-source, lo que favorece el desarrollo colaborativo mediante comunidades activas y la disponibilidad de numerosas bibliotecas para facilitar su programación. No obstante, a diferencia de las SBC, las SBM se emplean principalmente en aplicaciones simples debido a sus limitaciones de memoria, procesamiento y capacidades generales [29].

En cuanto a las SBC, se dispone de una variedad de modelos y marcas que van desde opciones básicas hasta sistemas de mayor capacidad. Esta diversidad facilita la elección de la alternativa más adecuada según las necesidades de cada proyecto. A continuación, se muestra una tabla con algunos ejemplos representativos [30].

2.4 Protocolos de Red

Según [15], un protocolo de red es un acuerdo técnico que establece cómo deben comunicarse los dispositivos dentro de una red, indicando el formato, sentido y secuencia de los mensajes. Además, define métodos de control, coordinación y tratamiento de fallos, asegurando interoperabilidad, confiabilidad y correcta interpretación entre equipos heterogéneos.

2.4.1 DHCP

DHCP, o Protocolo de Configuración Dinámica de Host, asigna de forma automática direcciones IP y parámetros de red a los dispositivos. Aumenta la facilidad de conexión, evita tener que proceder a la configuración manual además de asegurar que cada equipo tenga los datos que ha de tener para poder comunicarse en la red correctamente y optimizar así la gestión y la posibilidad de hacer uso de los recursos de red [1]

2.4.2 DNS

El sistema de nombres del dominio o DNS convierte los nombres de los sitios web en direcciones IP que comprenden las máquinas. Así, el usuario puede acceder a cualquier página web que quiera visitar sin recordar las complejas secuencias de números, haciendo que la localización de los recursos en internet sea mucho más sencilla y eficiente [1]

2.5 Protocolos de comunicación y gestión

En la comunicación de redes no basta con transmitir señales a través de un enlace físico, sino también es fundamental implementar mecanismos que permitan controlar y gestionar el intercambio de datos. Para ello, se incorpora una capa lógica sobre la capa física, conocida como control de enlace de datos, la cual garantiza una transmisión eficiente y confiable entre dispositivos conectados [15].

2.5.1 API (Application Programming Interface)

Una API es un conjunto de normas que permite a aplicaciones distintas interactuar y compartir datos o funciones de manera estructurada. Funciona como un intermediario que facilita solicitudes y respuestas, similar a un menú que guías pedidos sin revelar procesos internos [31].

2.5.2 API REST sobre HTTPS

La base fundamental de las aplicaciones web actuales son las APIs REST, ya que permiten un intercambio de información optimizado entre plataformas en entornos distribuidos y sobre infraestructuras en la nube. No obstante, estas pueden presentar vulnerabilidades de seguridad si no se implementan correctamente, como la transmisión de datos sin cifrado [32]. REST representa un estilo arquitectónico para sistemas que gestionan información multimedia interconectada en diferentes ubicaciones, promoviendo una interfaz uniforme en la web moderna [31].

Según [32], se propone el uso de HTTPS en APIs REST con el objetivo de mitigar amenazas comunes, como los ataques de tipo man-in-the-middle, la filtración de información y los accesos no autorizados. Además, se sugiere la implementación de métodos adicionales para fortalecer la seguridad de los datos, como el uso de TLS en sistemas estructurados, la verificación entre ambas partes, es decir, entre cliente y servidor, y la renovación periódica de los certificados digitales.

2.5.3 SSH

Destinado a ofrecer acceso remoto seguro a través de unas redes inseguras. Este protocolo está constituido por las tres partes más significativas: el protocolo de la capa de transporte es el que autentica la máquina del servidor y se encarga de ofrecer la integridad de los mensajes, el protocolo de autentificación del usuario se encarga de autenticar la máquina del cliente y también va a estar en la capa de transporte, el protocolo de conexión, que una vez autenticados los actores es el que divide el túnel cifrado generado en varios canales lógicos. SSH también soporta métodos de autenticación como el que está definido en el RFC 4256 (mensajes de intercambio genérico) o la interfaz GSS-API, se pueden generar sesiones seguras así tener el reenvío de puertos una vez efectuada la conexión establecida. [33].

2.5.4 VPN

En lo que respecta a las redes virtuales privadas (VPN), estas facilitan la conexión entre redes locales de manera más confiable, ya que utilizan canales públicos acompañados de mecanismos de cifrado y autenticación que garantizan la seguridad de la información transmitida [34]

Según [35], el uso de redes privadas virtuales (VPN) como herramienta de control remoto en el entorno industrial ofrece múltiples ventajas, entre las cuales destacan el cifrado de la

información, el acceso global, es decir, la posibilidad de conectarse desde cualquier ubicación y una baja latencia. Las características expuestas anteriormente hacen que, para proteger la propiedad intelectual en las fábricas, sean las VPN una opción adecuada para transmitir los comandos de forma dinámica y de manera segura.

2.6 Gestión de contenido

En el contexto de las redes domésticas, la gestión de contenidos ha llegado a cobrar una importancia notable, ya que forma una parte fundamental para que exista un entorno digital amparado y certero. Hoy en día se plantean y ofrecen un montón de métodos y tecnologías con este fin, el uso de servidores proxy, los cuales permiten filtrar y controlar el tráfico de la red, siendo uno de ellos. Esta gestión ha evolucionado desde listas negras con opciones muy básicas hasta sistemas mucho más elaborados y desarrollados, los cuales se alinean con la eficiencia, la escalabilidad y la adecuación a las necesidades de la casa moderna. [35].

2.6.1 Software de filtrado y monitoreo en redes domésticas

Hoy en día, los usuarios de las redes de un hogar suelen usar diferentes métodos para poder controlar, limitar o regular el uso de internet, principalmente en el ámbito de los niños y de los adolescentes. Hay diferentes aplicaciones o programas que ayudan a gestionar, manipular o controlar redes, mediante opciones como la tarea de filtrar tráfico, hacer el control del contenido y la seguridad del hogar. [36].

2.6.1.1 Pi hole

En cuanto a Pi-hole, este software funciona como un filtro DNS centralizado, lo que le permite actuar como un bloqueador de anuncios a nivel de red. A diferencia de otros bloqueadores tradicionales, Pi-hole opera directamente sobre el sistema de nombres de dominio (DNS), lo cual lo hace más eficaz al momento de restringir contenido no deseado y rastreadores [37].

Squid también permite bloquear publicidad identificando y anulando dominios de anuncios, filtrar sitios inapropiados con listas negras personalizadas, gestionar el sistema mediante una interfaz gráfica sencilla y funcionar en múltiples dispositivos, incluyendo equipos de bajo consumo como los basados en Raspberry Pi [37].

2.6.1.2 WireGuardian: Tailscale

WireGuard es una VPN moderna y de código abierto que prioriza la seguridad, velocidad y simplicidad en las conexiones. Desarrollada por Jason A. Donenfeld e integrada al kernel de Linux en 2020, destaca por su estructura ligera y el uso de criptografía avanzada como ChaCha20 y Curve25519, que aseguran una comunicación cifrada y eficiente [38]

Tailscale es un software que se basa en el motor VPN de WireGuard. Proporciona una capa de gestión que simplifica la creación de redes privadas seguras mediante la automatización de la autenticación, la administración de dispositivos y el establecimiento de conexiones cifradas entre ellos, es decir, reduce la complejidad de implementación y mantenimiento, lo que resulta adecuado para sistemas nuevos que no se acomplejan mucho [39].

2.6.1.3 Suricata

Al hablar de los sistemas de detección y prevención de intrusos (IDS/IPS), una de las herramientas más reconocidas es Suricata. Suricata es un sistema de código abierto diseñado para poder monitorizar el tráfico que hay en una red en tiempo real, utilizando un conjunto de reglas y firmas para poder detectar amenazas conocidas e identificar actividades sospechosas o anomalías de las comunicaciones de la red. Suricata también permite emitir alertas y registrar eventos de seguridad facilitando así el proceso de monitorización y análisis de los incidentes posibles. Gracias a su arquitectura de alto rendimiento y capacidad de procesamiento, Suricata puede implementarse tanto en redes domésticas como en entornos empresariales con grandes volúmenes de tráfico, lo

que lo convierte en una solución versátil y adecuada para proyectos orientados a fortalecer la seguridad y el monitoreo de redes [40].

2.6.2 Análisis heurístico

El análisis heurístico identifica malware al observar acciones anómalas usando reglas definidas y análisis en ambientes seguros. Se integra con inteligencia artificial para optimizar la detección, facilitando el reconocimiento de amenazas inéditas y minimizando errores en resultados. Sus principales características incluyen el examen de código estático y dinámico en entornos controlados, la comparación con bases de datos conocidas, y la simulación de la ejecución para detectar comportamientos maliciosos, lo que contribuye a evitar falsos positivos [41].

2.6.2.1 Técnicas de análisis heurísticas

Dentro del ámbito de la heurística, se emplean diversas técnicas que permiten optimizar las reglas utilizadas para la detección de actividades sospechosas en una red. En la siguiente tabla se puede observar las técnicas heurísticas:

Tabla 1

Técnicas de análisis heurístico

Técnica	Descripción
Análisis estático	• Analiza el código sin ejecutarlo, es rápido y seguro, sin embargo, puede generar falsos positivos y no detectar comportamientos que no están definidos.

Análisis dinámico	• Ejecuta el archivo en un entorno aislado para observar su comportamiento real, detectando acciones maliciosas, aunque requiere mayor tiempo y recursos.
Monitoreo de tráfico	• Tiende a examinar los diferentes flujos de red, con la finalidad de lograr identificar las diferentes anomalías en tiempo real, pero para poder realizarlo, depende de que exista una supervisión de forma continua y con la presencia de una correcta configuración.

Nota: *Métodos complementarios usados para detectar comportamientos maliciosos en sistemas.*

2.6.2.2 Limitaciones del análisis heurístico

Uno de los aspectos más negativos del análisis heurístico está en su apertura a generar falsos positivos, esto es, a detectar erróneamente a archivos válidos como maliciosos. Para limitar esta debilidad, el análisis heurístico se utiliza, por lo general, complementado con el método de firmas, el cual proporciona por lo general una mayor precisión y una mayor reducción de errores, sobre todo frente a amenazas nuevas [42].

Capítulo III: Diseño e Implementación

En el presente capítulo se expone el desarrollo del prototipo, el cual se aplica bajo la metodología cascada, lo que permite estructurar las fases de diseño, implementación y validación de forma secuencial. Asimismo, se analizan los requerimientos necesarios para garantizar una implementación eficiente del sistema. Se incluye el proceso de recolección de información mediante encuestas dirigidas a los usuarios, con el propósito de definir adecuadamente las necesidades y funciones del sistema. Finalmente, se describen los procedimientos realizados conforme a la metodología seleccionada y los resultados obtenidos.

3.1 Metodología

El desarrollo del prototipo se realizó bajo el modelo en cascada, debido a que permite organizar las fases de forma secuencial y ordenada. Cada etapa debía completarse antes de iniciar la siguiente, lo que garantizó un proceso controlado y una documentación clara [6]. Las fases representan desde los requerimientos hasta las pruebas necesarias para cumplir con el proyecto.

- **Análisis de Requerimientos**: En esta fase se recopila y documenta toda la información necesaria para comprender el problema y establecer los objetivos del sistema.
- **Diseño del sistema:** Se definen la arquitectura general y los componentes del sistema, especificando cómo interactuarán entre sí para cumplir con los requerimientos.
- **Implementación**: Se ejecuta la construcción del sistema, desarrollando la parte software y configurando la parte hardware, siguiendo el diseño que ha quedado acordado anteriormente.

- **Pruebas**: El sistema se somete a pruebas de funcionalidad y de rendimiento, para verificar que funciona correctamente y que se han cumplido los objetivos que se propusieron.

3.2 Análisis de Requerimientos

En esta etapa se pretende explorar la determinación de los requerimientos más significativos del proyecto, ya sea desde el punto de vista del software con su correspondiente programación y su lógica interna o tomando en cuenta el hardware que permita un rendimiento del sistema adecuado al problema que se está resolviendo. A su vez, también debe tenerse en cuenta aquellos parámetros a considerar para plantear la solución más óptima, a partir del problema existente y estudios similares para elevar el diseño de optimización. Por último, se detallan los errores y los problemas posibles que pueden sobrevenir del desarrollo que permita su previsión y gestión adecuadas.

3.2.1 Situación Actual

Si en la actualidad la conectividad a internet ha aumentado con gran fuerza en los hogares, con el resultado de que los niños y los adolescentes puedan acceder a información, entretenimiento y educación, a la vez se ven expuestos a contenidos maliciosos e inadecuados. La pandemia incrementó notablemente el uso de dispositivos móviles por parte de los niños debido a la educación a distancia, a la realización de tareas, a la comunicación continua con los educadores y con otros compañeros, propiciando la aceleración del uso de la tecnología en el ámbito de la educación. Este paso repentino de lo presencial anteriormente a lo digital ha generado un fenómeno donde muchos niños permanecen pegados a las pantallas como fenómeno de que los niños vean reducidos el tiempo de juego y el tiempo de socialización en la vida real. [4]

Aunque existen diversas soluciones de software de control parental, muchas presentan limitaciones como configuraciones complejas y cobertura insuficiente frente a nuevas amenazas [43].

Ante esta situación, se planteó un prototipo basado en una computadora de placa única (SBC), que implementa filtrado mediante listas negras, control de tiempo de uso del internet, reportes y análisis heurístico para detectar amenazas emergentes. Todo esto con ayuda de una aplicación móvil que ayudara a gestionar la red.

A diferencia de los controles parentales convencionales, este enfoque no solo protege frente a contenidos inapropiados a todos los equipos que hagan uso de su red, sino que también promueve un uso responsable de la tecnología, limitando el acceso a recursos digitales para incentivar actividades fuera de pantalla, fomentando un equilibrio entre seguridad, entretenimiento y desarrollo saludable.

3.2.2 Estudio de la Población Objetivo

El presente estudio se desarrolló en la ciudad de Cotacachi, específicamente en la parroquia El Sagrario, y estuvo dirigido a docentes de escuelas de educación básica. El estudio se centró en esta población debido a su intervención directa en la formación de los niños y en su vínculo con los padres, quienes son la población objetivo del proyecto. A fin de delimitar el alcance del estudio y para asegurar su viabilidad, se optó por abarcar únicamente las instituciones educativas del sector urbano, dejando fuera a los centros educativos que impartieran una cobertura sólo rural o sólo de educación secundaria.

$$\mathrm{n} = \frac{N * Z^2 * p * q}{e^2(N-1) + Z^2 * p * q} \qquad (1)$$

$$n = \frac{92 * (1.96)^2 * 0.5 * 0.5}{(0.10)^2(92-1) + (1.96)^2 * 0.5 * 0.5} = 47.24$$

Para determinar el tamaño de la muestra se utilizó la fórmula para poblaciones finitas que se sitúa en la Ecuación (1), contemplando en este caso un total poblacional de 92 docentes. Se ha determinado un nivel de confianza del 95%, que corresponde a un valor Z de 1.96; una probabilidad de que ocurra lo que se investiga, p = 0.5 y q= 0.5 y también un margen de error del 10%.

Una vez sustituido los valores en la fórmula e igualado, se obtuvo un tamaño para la muestra de 47.24 participantes. Siguiendo el criterio de la investigación científica, este valor se redondeó al número entero más cercano, concluyéndose así una muestra de 47 docentes, el número de docentes que se consideran adecuados y representativos para la fundamentación del estudio. En el anexo 2 se describe la encuesta en general y los resultados.

3.2.3 Propósito del sistema

El fin que persigue la propuesta de implementar un prototipo de control parental es el de implementar un sistema tecnológico que ayude a los padres y a los tutores a gestionar de una manera segura el acceso a Internet en el hogar. Esta propuesta de sistema va dirigida a poder proteger a los menores de los contenidos maliciosos o inapropiados y conseguir un entorno digital fiable, regulado y con facilidad de gestión.

Este prototipo podrá ser utilizado en contextos de red doméstica, actuando como nodo intermedio entre la conexión de los dispositivos con acceso a Internet, y el acceso a Internet. Sus funcionalidades abarcan filtrado de tráfico mediante listas negras, análisis heurístico para la

detección de amenazas emergentes, control de horarios asociados al uso de internet, generación de reportes de actividad y una interfaz de administración disponible a través de una aplicación móvil, orientada a la facilidad de uso para usuarios sin conocimientos técnicos avanzados.

Los beneficiarios directos del prototipo son los padres, madres y tutores legales de menores de edad, quienes contarán con una solución tecnológica que facilite la supervisión y regulación del acceso a internet en el hogar. De forma indirecta, los menores se constituyen en usuarios protegidos, al disminuir su exposición a amenazas digitales como contenido inapropiado, malware o ciberacoso, favoreciendo así un entorno de navegación más seguro y promoviendo un uso responsable de las tecnologías de la información en el ámbito familiar. De esta manera, se garantiza un entorno digital seguro y optimo en el hogar.

3.2.4 Trabajos relacionados

Para comprender adecuadamente el alcance del proyecto, es necesario tener en cuenta diversos aspectos esenciales que permitan satisfacer los requerimientos del prototipo de control parental. En este sentido, se analizan estudios relacionados que abordan temas relevantes como la seguridad de la red, la protección de datos y los mecanismos de filtrado de información todo esto en el anexo 1.

En el estudio relacionado 1, expone el diseño e implementación de un sistema de monitoreo y control parental en redes locales, basado en software libre y ejecutado en una computadora de placa única como Raspberry Pi. Su finalidad es reducir el acceso de menores a contenido inapropiado, permitiendo a los tutores supervisar, restringir y gestionar la navegación web mediante filtros, reportes y reglas personalizadas. Asimismo, analiza riesgos digitales y técnicas

de bloqueo empleando herramientas como proxy, SquidGuard, DNS y SARG, proponiendo una alternativa económica y funcional.

En el estudio relacionado 2, consiste en el desarrollo de un sistema de seguridad para redes domésticas usando un Firewall de Nueva Generación orientado a proteger a niños durante su navegación en Internet. La solución incorpora políticas avanzadas de control, permisos por usuario, monitoreo constante y filtrado de contenido y aplicaciones. Utiliza un modelo iterativo para perfeccionar el sistema que se está desarrollando, añadiendo así un Access Point con VLAN y un MiniPC x86_64, así como funciones de prevención de intrusos, detección de malware, generación de informes completos y una interfaz web para la gestión.

En el trabajo vinculado 3 se plantea un diseño e instalación de un Firewall-Proxy de GNU/Linux para hacer un mejor uso de los recursos técnico-tecnológicos en la Facultad de Ingeniería de la Universidad Técnica del Norte. Su propósito es mejorar el rendimiento del Data Center mediante filtrado, control y administración del tráfico, utilizando herramientas libres como Squid, iptables y Webmin. Además, plantea políticas de seguridad, segmentación en VLAN y pruebas funcionales para validar su impacto en eficiencia y protección.

En el estudio relacionado 4, plantea un sistema avanzado de control parental para Smart TVs mediante el uso de televisores rooteados, lo que permite acceder a funciones que el fabricante restringe. El sistema habilita monitoreo remoto, control total del dispositivo y gestión extendida del contenido al que acceden los usuarios, superando ampliamente los mecanismos básicos de bloqueo por PIN. Su propósito es ofrecer una plataforma independiente del fabricante, capaz de funcionar incluso en televisores antiguos, permitiendo un control parental más completo, flexible y orientado a la seguridad digital en el hogar.

En el estudio relacionado 5, propone una herramienta llamada HomeShark, desarrollada en Python, para monitorear el tráfico de Internet de todos los dispositivos de una red doméstica desde un solo equipo. A diferencia de soluciones comerciales costosas o herramientas técnicas como Wireshark, la aplicación presenta la información de forma comprensible para padres o tutores. Utiliza consultas DNS para identificar sitios y servicios utilizados, permitiendo conocer la actividad en línea sin instalar software en cada dispositivo, facilitando así un control accesible y centralizado.

3.3 Requerimientos

Esta sección tiene como propósito establecer la relación entre los objetivos planteados y las funcionalidades que el sistema debe garantizar durante su desarrollo. Para tal fin, se sitúan los requerimientos elaborados tanto a partir de las necesidades que se han recogido de los usuarios, mediante la aplicación de las encuestas, como de los criterios técnicos definidos para el diseño del prototipo.

El análisis de requerimientos se realiza a partir de las directrices del estándar ISO/IEC/IEEE 29148:2018, dado que, este estándar proporciona un marco de actuación sistemático para la identificación, la documentación y la validación de los requisitos del sistema. Este análisis de los requerimientos permite organizar las funciones y las condiciones necesarias para garantizar la calidad, la seguridad y la usabilidad del control parental y validar que se cumplan las necesidades de los beneficiarios en los objetivos técnicos del proyecto.

3.3.1 Stakeholders

Los stakeholders, o partes interesadas, son las personas o grupos que se relacionan con un sistema, ya sea por su uso (directo), por su gestión o por la influencia que este tiene en sus tareas. En el diseño del prototipo de control parental, es fundamental identificar a los actores implicados

y averiguar cuáles son las necesidades de estos. Cada grupo requiere unos requisitos que han de ser considerados durante la construcción y explotación del sistema, para garantizar su utilidad, fácil manejo y fidelidad en el tiempo.

Antes de definir las especificaciones funcionales del sistema, es esencial identificar y analizar a los diferentes stakeholders involucrados, así como sus necesidades y aportes específicos al desarrollo del proyecto. Este análisis permite establecer una base sólida para la definición de los requerimientos, garantizando que el sistema cumpla con las expectativas tanto técnicas como funcionales de los participantes.

Tabla 2

Actores involucrados

Actores Interesados		
N°	**Actor**	**Función**
1	Bryan Arotingo	Estudiante
2	Msc. Fabián Cuzme	Director
3	Msc. Carlos Vásquez	Asesor

Nota*: Esta tabla identifica los actores involucrados dentro del proyecto.*

3.3.2 Nomenclatura de requerimientos

Con el fin de garantizar un control eficiente durante el proceso de desarrollo, es necesario asignar una nomenclatura identificadora a cada requerimiento. En la siguiente tabla, se exponen las abreviaturas normalizadas empleadas para clasificar y gestionar los distintos tipos de requerimientos definidos en este proyecto, asegurando coherencia con las directrices establecidas por el estándar ISO/IEC/IEEE 29148:2018.

Tabla 3

Nomenclatura de los requerimientos

Descripción	Abreviatura
Requerimientos de Stakeholders	StSR
Requerimientos de Sistema	SySR
Requerimientos de Arquitectura	SrSH

Nota: *En esta tabla se detalla las abreviaturas de cada requerimiento necesario para el desarrollo del proyecto.*

Dentro de la fase de levantamiento y análisis de requerimientos, se procede a definir cada uno de ellos con el fin de identificar su relevancia dentro del sistema. Para ello, se consideran los recursos y medios disponibles, estableciendo un nivel de prioridad que permita determinar el nivel de impacto y urgencia en su implementación. Las prioridades se clasifican en alta, media y baja, y se detallan en la siguiente tabla.

Tabla 4

Prioridades de los requerimientos del prototipo de control parental

Prioridad	Descripción
Alta	Es un requerimiento indispensable y crítico para el sistema, cuya ausencia afectaría su funcionamiento y objetivos, por lo que debe asegurarse y cumplirse durante todas las etapas del desarrollo.
Media	Este requerimiento puede descartarse únicamente ante circunstancias excepcionales, analizando impactos y riesgos, y dejando constancia formal con aprobación del equipo responsable y las partes involucradas.

Baja	Este requerimiento puede omitirse sin afectar el funcionamiento general, siempre que la decisión sea analizada, justificada, documentada y aprobada por el equipo y los interesados correspondientes.

3.3.3 Requerimientos de stakeholders

Los requisitos de los stakeholders representan las necesidades, objetivos, expectativas y condiciones planteadas por las partes interesadas que el sistema debe satisfacer. Estos requisitos reflejan lo que cada actor espera del funcionamiento del sistema y orientan su diseño inicial. En la siguiente tabla se presentan los requisitos identificados como necesarios para cumplir con dichas demandas.

Tabla 5

Requerimientos de stakeholders

Requerimientos de stakeholders (StSR)				
N°	**Requerimientos**	**Prioridad**		
		Alta	**Media**	**Baja**
Requerimientos operacionales				
StSR1	El sistema deberá funcionar de manera continua para garantizar filtrado y supervisión en tiempo real de la red doméstica.	X		
StSR2	El sistema deberá permitir autorización para administrar todos los dispositivos conectados al punto de acceso del SBC.	X		
StSR3	El sistema deberá permitir bloquear dominios y Ads maliciosos o inapropiados	X		
StSR4	El sistema deberá integrar análisis heurístico para detectar comportamientos anómalos en la navegación cotidiana.	X		

StSR5	El sistema deberá permitir conexión inalámbrica (WiFi) estable para los usuarios finales sin afectar su rutina diaria.	X		
StSR6	El sistema deberá enviar alertas inmediatas ante intentos de acceso a contenido inapropiado.		X	
StSR7	El sistema deberá generar reportes detallados de actividad de navegación		X	

Requerimientos de usuarios

StSR8	El sistema deberá proporcionar una interfaz intuitiva y fácil de usar para personas sin conocimientos técnicos.	X		
StSR9	El sistema deberá permitir la definición de horarios para limitar el acceso de los menores a internet.	X		
StSR10	El sistema deberá permitir excepciones personalizadas según los dispositivos autorizados.		X	
StSR11	El sistema debe permitir visualizar estadísticas y reportes comprensibles sobre la actividad de la red.		X	
StSR12	El sistema deberá permitir utilizarse desde fuera del hogar.	X		
StSR13	El sistema deberá ser compacto y operar con un consumo energético reducido para facilitar su instalación y uso continuo en el hogar.		X	

3.3.4 Requerimientos del sistema

Antes de iniciar el diseño funcional del sistema, resulta necesario identificar y definir los requerimientos que lo conforman. Los requerimientos del sistema definen formalmente funciones, restricciones y condiciones que debe cumplir, garantizando necesidades de las partes interesadas y sirviendo como base verificable para diseño, desarrollo y validación del proyecto.

Tabla 6

Tabla de Requerimientos del sistema

Requerimientos del Sistema (SyRS)					
N°	**Requerimientos**	**Prioridad**			**Relación**
		Alta	**Media**	**Baja**	
Requerimientos funcionales					
SySR1	El sistema debe ser intuitivo para los usuarios finales.	X			**StSR8**
SySR2	El sistema deberá aplicar filtrado DNS para bloquear dominios y Ads clasificados como inapropiados.	X			**StSR3**
SySR3	El sistema deberá ejecutar un sistema de IDS para inspeccionar tráfico web.	X			**StSR1**
SySR4	El sistema deberá permitir configurar horarios de acceso a Internet.	X			**StSR9**
SySR5	El sistema deberá generar logs detallados del tráfico para reportes y alertas.		X		**StSR7**
SySR6	El sistema deberá permitir control remoto desde la aplicación móvil mediante API.	X			**StSR8, StSR12**
Requerimientos No funcionales					
SySR7	El sistema deberá operar de forma continua (24/7) sin interrupciones.	X			**StSR1**
SySR8	El sistema deberá ser compacto y de bajo consumo energético para instalación doméstica.		X		**StSR13**
SySR9	La interfaz móvil deberá ser clara, sencilla y apta para usuarios sin conocimientos técnicos.		X		**StSR8, SySR1**

SySR10	El sistema deberá soportar al menos 10 dispositivos conectados simultáneamente			X	**StSR5**
Requerimientos de Seguridad					
SySR11	El sistema deberá cifrar la comunicación entre la app y el SBC mediante VPN.	X			
SySR12	El sistema deberá exigir autenticación segura para acceder a la aplicación móvil.	X			
SySR13	El sistema deberá impedir que los menores modifiquen configuraciones de la aplicación.	X			
Requerimientos de Interfaz					
SySR14	El sistema debe implementarse utilizando software de código abierto.	X			
SySR15	El sistema debe permitir observar las actividades y reportes.		X		**StSR11**
SySR16	El sistema debe tener una app para una mejor experiencia.	X			**StSR8, SySR1**
SySR17	El sistema debe ser compatible para el sistema operativo Android.	X			**SySR14**
Requerimientos Físicos					
SySR18	El sistema debe ser compacto y permitir una manipulación sencilla.		X		**SySR8, StSR13**
SySR19	El sistema debe ser óptimo en ambientes acomplejados		X		

3.3.5 Requerimientos de la arquitectura

Los requerimientos de arquitectura describen la organización interna del sistema y la forma en que se integran sus distintos elementos físicos y lógicos. Estos lineamientos consideran aspectos relacionados con la estructura del hardware, el comportamiento del software y las conexiones eléctricas necesarias para su operación. Su propósito es asegurar que el sistema funcione de manera coherente, ordenada y sostenible. Para ello, se analizan criterios de diseño, distribución de módulos, interacción entre componentes y necesidades energéticas. En la siguiente tabla se detalla los requerimientos de la arquitectura.

Tabla 7

Requerimientos de la Arquitectura

Requerimientos de Arquitectura (SrSH)					
N°	**Requerimientos**	**Prioridad**			**Relación**
		Alta	**Media**	**Baja**	
Requerimientos operacionales					
SrSH1	El sistema deberá tener la capacidad de supervisar en tiempo real la actividad del tráfico dentro de la red local.	X			**StSR**4, **StSR7**
StSH2	El sistema debe ser capaz de analizar el tráfico mediante servidores DNS y IDS.	X			**SySR2, SySR3**
StSH3	El sistema deberá realizar la autenticación del usuario a través de la aplicación móvil.	X			**StSR2**
StSH4	El sistema debe analizar el tráfico general de la red	X			**StSR4**
Requerimientos de Diseño					
StSH5	El sistema debe ser capaz de entregar una señal eficiente mediante WiFi 2.4 Ghz.	X			**StSR5**

StSH6	El sistema deberá disponer de un puerto destinado al acceso y conexión con la red WAN.	X			
StSH7	Memory RAM: El sistema debe contar con 8 Gb como mejor opción.		X		
StSH8	Procesador: Debe tener un procesador de 4 núcleos ARM v8-A 64bits.	X			
StSH9	ROM: Debe contar con un mínimo de 128 Gb para almacenamiento de datos.		X		
Requerimientos de Software (SHRS)					
N°	**Requerimientos**	**Prioridad**			**Relación**
		Alta	**Media**	**Baja**	
Requerimientos Lógicos					
SHRS1	El sistema debe permitir el inicio de sesión seguro para padres o tutores mediante usuario y contraseña.	X			**StSH3**
SHRS2	El sistema debe analizar logs de actividad de los menores.	X			**StSH4**
SHRS3	El sistema debe garantizar la integridad y confidencialidad de los datos almacenados.	X			
SHRS4	El sistema debe ofrecer una interfaz lógica clara para la configuración y monitoreo.	X			**SySR16**
SHRS5	El sistema debe ser integrable con herramientas open source.	X			**SySR14**
SHRS6	El sistema debe ser escalable y flexible.		X		
Requerimientos de Software					

SHRS7	El software para el sistema debe ser de código abierto o de libre uso.	X			**SySR14**
SHRS8	El sistema debe permitir la integración con aplicaciones desarrolladas en Python.	X			
SHRS9	El software deberá garantizar un funcionamiento seguro y eficiente durante toda la operación del sistema.	X			**StSR1**
SHR10	Sistema operativo compatible con software open source.	X			
SHRS11	El software debe ser flexible y adaptable a las necesidades y preferencias del usuario.		X		
SHRS12	El sistema debe tener un backend y frontend	X			
Requerimientos de Hardware					
SHRS12	El sistema debe contar con una alimentación adecuada para un buen funcionamiento.	X			
SHRS13	El sistema debe ser tener un almacenamiento mínimo a 128 GB.	X			**StSH9**
SHRS14	El sistema debe tener dos interfaces de red, Inalámbrica y cableada.	X			**StSH5**
SHRS15	El sistema debe tener un sistema de enfriamiento.		X		
SHRS16	El sistema debe estar con una base de 4GB en RAM.	X			**StSH7**
SHRS17	El sistema deberá contar con una protección física resistente que minimice daños por golpes o impactos externos.	X			**SySR19**

3.4 Selección de Hardware y software

La elección del hardware y software se formalizó a partir de un análisis comparado de distintas alternativas, en el que se explicitaron sus especificaciones técnicas, en relación con las características del proyecto. Este tipo de procedimiento ayudó a sortear la elección basada en la experiencia previa y asegura que cada uno de los componentes involucrados proporcione la funcionalidad y el apoyo adecuado a la operación de red.

La evaluación se fundamenta en el grado en que las diferentes propuestas elegidas cubren los requerimientos ya definidos (StSR), los requerimientos funcionales y no funcionales (SySR), así como las pautas arquitectónicas (SrSH). A través de una matriz de validación se obtuvieron las opciones más adecuadas, de las que se descartaron las que presentaban limitaciones de rendimiento o de compatibilidad, facilitando así la elección de una solución escalable y eficiente.

3.4.1 Hardware

Una vez determinado los requerimientos y realizado un análisis detallado de cada componente, se determinó la implementación de un sistema optimizado, con soporte activo y compatibilidad con programas estables y eficientes. Por ello, se seleccionó una Raspberry Pi 5 como equipo principal. En esta sección se llevará a cabo el análisis del hardware necesario, considerando distintas placas y sus características técnicas. A partir de los requerimientos establecidos, se realizará una comparación que permitirá identificar la opción más adecuada para el desarrollo del sistema.

Tabla 8

Requerimientos de SBC

Single Board Computer				
Requisitos	**Orange Pi 5+**	**Google Coral Dev Board**	**Raspberry Pi 5**	**Nvidia Jetson Nano**
SySR7	1	1	1	1
SySR18	1	1	1	1
SySR19	1	0	1	1
StSH5	0	0	1	0
StSH6	1	1	1	1
StSH7	1	0	1	1
StSH8	1	0	1	1
StSH9	1	0	1	1
StSH13	1	0	1	1
StSH14	1	0	0	0
StSH15	0	0	1	1
Total	**9**	**3**	**10**	**9**

1 = Cumple, 0 = No cumple

A partir de los resultados obtenidos en la Tabla 8, se observa que la Raspberry Pi 5 alcanza la mayor puntuación con 10 puntos, superando a las demás opciones evaluadas. Este resultado evidencia que cumple de manera más completa los requerimientos establecidos, destacándose por

su compatibilidad, soporte de hardware y facilidad de integración con los demás componentes del sistema.

3.4.1.1 Selección de Hardware

Por estas razones, la Raspberry Pi 5 se considera la opción más adecuada para la implementación del hardware del proyecto.

A continuación, se presentan sus características técnicas y la justificación de su elección en función de los requerimientos establecidos.

Tabla 9

Hardware Seleccionado

RASPBERRY PI		
		
Parámetro	**Características de la Pi 5**	**Evaluación técnica**
Sistema Operativo	Raspberry Pi OS	Soporte para entornos de procesos grandes y IoT.
CPU / GPU	Broadcom BCM2712 Quad-core Arm Cortex-A76 a 2,4 GHz con una GPU VideoCore VII 800 MHz.	Garantiza el trabajo múltiple de servicios ayudado con su GPU.
RAM	8 GB LPDDR4X-4267	Permite un sistema más fluido para procesos grandes.

Almacenamiento	Compatible con microSD SDR104 en este caso con de 128 GB.	Cumple la velocidad requerida, con opción de adaptar una unidad SSD.
Interfaz de red	Puerto Gb Ethernet y Comunicación inalámbrica Wifi 2.4/5 Ghz.	Conexión por cable y red inalámbrica de alta velocidad.
Consumo energético	Alta carga hasta 27 W	Puede llegar a sobrecalentarse si no hay un sistema de ventilación adecuada

La Raspberry Pi 5 cumple de manera adecuada con los requerimientos técnicos del proyecto, permitiendo la ejecución simultánea de múltiples servicios como filtrado DNS, IDS, base de datos y API sin afectar el rendimiento del sistema. Además, su capacidad de expansión, conectividad de red y compatibilidad con sistemas operativos orientados a IoT la convierten en una plataforma confiable y escalable. Aunque su consumo energético puede aumentar bajo alta carga, este aspecto puede ser controlado mediante un sistema de ventilación adecuado, garantizando así un funcionamiento estable y continuo.

3.4.2 Software

En esta parte al igual que en hardware, se analizará el software necesario para el sistema, considerando una evaluación comparativa basada en los requerimientos previamente establecidos. Con esta ponderación se puede llegar a mencionar que será posible llegar a determinar la opción más idónea al momento de llegar a su implementación.

3.4.2.1 Selección del servidor DNS

La sección presente aborda las alternativas de software para la implementación del servidor DNS que cumplen con unos requisitos funcionales, de seguridad y de compatibilidad del software con el propuesto en el capítulo anterior.

Gracias a la evaluación comparativa se permite poder determinar la opción que mejor se acomoda a los requisitos establecidos y asegurar un adecuado control de la resolución de nombres, filtrado de contenido y respuesta a la administración de red.

Tabla 10

Requerimientos para el DNS

	Servidor DNS			
Requisitos	**Pihole**	**Adguard Home**	**NextDNS**	**OpenDNS**
StSR3	1	1	1	0
StSR7	1	1	0	0
StSR11	1	1	1	1
SySR2	1	1	1	1
SHRS7	1	1	0	1
SHRS11	1	0	0	1
Total	**6**	**5**	**3**	**4**
1 = Cumple, 0 = No cumple				

De acuerdo con los resultados obtenidos en la Tabla 10, Pi-hole y AdGuard Home alcanzan la mayor puntuación con 6 y 5 puntos respectivamente, superando a NextDNS y OpenDNS. Sin embargo, Pi-hole destaca por cumplir la totalidad de los requerimientos funcionales y de seguridad evaluados, además de ser una solución de código abierto que permite una mayor personalización e integración con otros componentes del sistema. Por estas razones, Pi-hole se selecciona como el servidor DNS más adecuado para el proyecto.

3.4.2.2 Selección de la base de datos

En esta tabla se presenta la evaluación de los sistemas gestores de bases de datos considerados para el proyecto. El análisis se basa en criterios de rendimiento, escalabilidad, compatibilidad, seguridad y soporte para aplicaciones web. La selección de la base de datos adecuada es fundamental para garantizar la correcta gestión, almacenamiento y disponibilidad de la información generada por el sistema.

Tabla 11

Requerimientos de Bases de datos

	Servidor Base de datos			
Requisitos	**Mysql**	**MariaDB**	**Postgresql**	**MongoDB**
SySR7	1	1	1	1
SySR12	1	1	1	1
SySR14	1	1	1	1

SySR17	1	1	1	1
SHRS3	1	1	1	1
SHR11	0	1	0	0
Total	**5**	**6**	**5**	**5**
1 = Cumple, 0 = No cumple				

Según la Tabla 11, MariaDB obtiene la puntuación más alta con 6 puntos, mientras que MySQL, PostgreSQL y MongoDB alcanzan 5 puntos. SQLite se selecciona debido a su ligereza, facilidad de implementación y cumplimiento total de los requerimientos definidos, lo que la convierte en una opción adecuada para sistemas con requerimientos moderados de almacenamiento y bajo consumo de recursos, como el propuesto en este proyecto.

3.4.2.3 Selección de Framework para Backend

En el presente apartado se comparan distintos frameworks utilizados para el desarrollo de la API del sistema. La evaluación considera aspectos como rendimiento, facilidad de desarrollo, seguridad y soporte para servicios REST. Esta comparación permite identificar el framework más adecuado para asegurar una comunicación eficiente y segura entre los distintos componentes del sistema.

Tabla 12

Requerimientos de API

Requisitos	FLASK	FastAPI	Django
SySR6	1	1	0
SySR7	1	1	1

SHRS5	1	1	1
SHRS7	1	1	1
SHRS11	0	1	1
Total	**4**	**5**	**4**
1 = Cumple, 0 = No cumple			

De acuerdo con los resultados presentados en la Tabla 13, FastAPI obtiene la puntuación más alta con 5 puntos, superando a FLASK y Django. FastAPI se selecciona debido a su alto rendimiento, soporte nativo para APIs REST y sus diferentes librerias, validación automática de datos y documentación integrada, características que garantizan una comunicación eficiente, segura y escalable entre los componentes del sistema.

3.4.2.4 Selección de Framekwork para Frontend

Finalmente, se analizan las plataformas de desarrollo para la aplicación móvil, considerando criterios como compatibilidad, facilidad de desarrollo, soporte multiplataforma y escalabilidad. La evaluación comparativa permite seleccionar la herramienta que mejor se ajusta a los requerimientos funcionales del proyecto y a las necesidades de los usuarios finales.

Tabla 13

Requerimientos para la aplicación móvil

Requisitos	**MIT App Inventor**	**Android Studio**	**React Native**
StSR8	0	1	1
StSR10	1	1	1

SySR7	1	1	1
SHRS7	1	1	1
SHRS8	0	1	1
Total	3	5	5
1 = Cumple, 0 = No cumple			

En la Tabla 14 se observa que React Native y Android Studio obtiene la mayor puntuación con 5 puntos, superando a MIT App Inventor. Estas dos herramientas se seleccionan una por su facilidad de desarrollo, rápida implementación y cumplimiento total de los requerimientos funcionales definidos, lo que la convierte en una opción adecuada para prototipos funcionales y aplicaciones orientadas a usuarios finales sin alta complejidad técnica y con compatibilidad en SO. Otra por su amplia librería y compatibilidad con el SO de Android puro, pero esta solo se usará para se sirva de recolector de información.

3.4.3 Software Seleccionado para el Sistema

El paquete de software fue seleccionado por sus características: su eficiencia, su bajo consumo y su facilidad de integración en el sistema del control parental que se ha creado. Las herramientas de software permiten: la gestión del filtrado de contenido, la administración y asignación de red, el control del tráfico web y la implementación de almacenamiento ligero sin necesidad de servidores pesados y de gran complejidad. También la comunicación existente entre módulos de software es bastante sencilla y, la interfaz existente para dispositivos móviles permite la administración remota de una forma bastante intuitiva y segura.

Tabla 14

Software seleccionado

Servicio	Funciones	Características	Integración al sistema
Pi-hole	Filtrado de publicidad y bloqueo de dominios.	Pi-hole bloquea anuncios con DNS interno y listas configurables (Blacklist).	Gestiona peticiones DNS y bloqueos de red. Solo actuara como server DNS no como DHCP.
dnsmasq	DHCP y resolución DNS local.	Gestión de DNS, DHCP y TFTP, optimizando redes locales eficientemente.	Asigna IPs a clientes del AP. Pero no se usa como server DNS.
MAriaDB	Base de datos local.	Ligera, formato .db, sin servidor.	Guarda usuarios y configuraciones.
Python FastAPI (API)	Backend de comunicación.	REST, integración con MariaDB.	Conecta la app con el sistema.
React Native	Interfaz móvil de control parental.	Conexión HTTP, entorno Android e IOS de forma nativa.	Gestión y administración total para el administrador mediante API y Childguard.
Android Studio	Interfaz móvil de Childguard	Conexión HTTP, entorno Android.	Monitoreo y accesibilidad del dispositivo cliente.

El software que se eligió fue fundamentalmente el que se caracteriza por su eficacia, escaso consumo de recursos y facilidad de integración dentro del sistema de control parental. Estas herramientas proveen funciones esenciales como la filtración de contenido a nivel de DNS,

filtración de direcciones IP, control del tráfico web y almacenamiento de la información de forma ligera sin necesidad de servidores monstruosamente pesados. La comunicación entre los módulos del sistema se ejecuta de forma simple mediante un API y la aplicación móvil permite operar el sistema de control parental de forma intuitiva y a distancia.

3.5 Diseño del Sistema

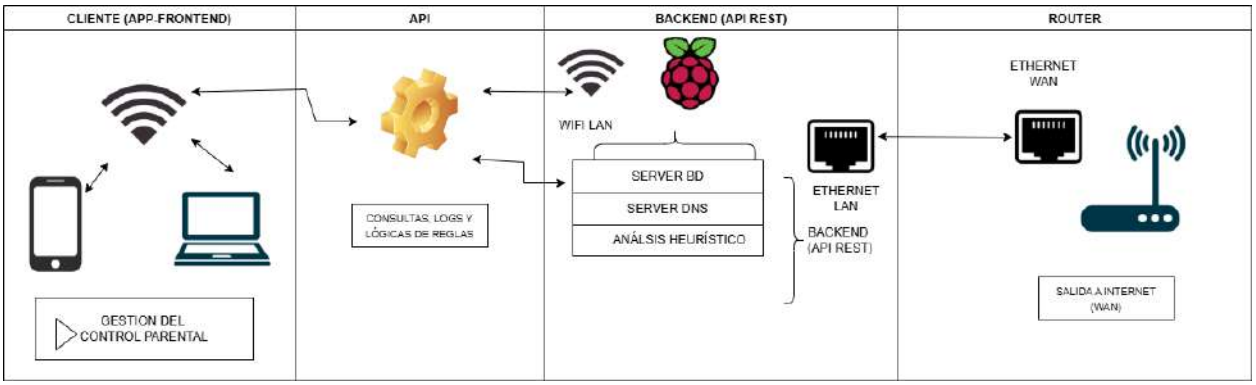
En esta sección se presenta el diseño conceptual y estructural del sistema de control parental inteligente, desarrollado sobre una Raspberry Pi 5 que opera como gateway de red. El diseño del prototipo de control parental se fundamentó en la selección de herramientas de código abierto y hardware eficiente. Además, se presentan diagramas que abordan una perspectiva distinta del mismo sistema, de modo que en conjunto permiten comprender tanto su estructura estática como su comportamiento dinámico.

3.5.1 Arquitectura del Sistema

Una vez establecidos los requisitos y determinados los elementos de hardware y software que conformarán el sistema, es necesario estructurar con claridad su organización interna. Para ello, se procede a definir el modelo arquitectónico que describe cómo interactúan los distintos módulos. Esta organización permite que cada componente pueda evolucionar o sustituirse sin afectar directamente a las demás capas, siempre que se respeten las interfaces de comunicación entre ellas.

Figura 5

Arquitectura del control parental



Nota: Esta ilustración se demuestra la conexión de los dispositivos de red.

Capa Cliente (Frontend)

En la capa Cliente se concentra los puntos de interacción con el usuario final es decir la aplicación móvil desarrollada en React Native, el sistema de notificaciones push gestionado mediante Expo Notifications, y de forma complementaria el panel administrativo web nativo de Pi-hole, utilizado durante el desarrollo para tareas de diagnóstico.

Capa API (Interfaz de Comunicación)

La capa API, implementada a través de FastAPI, gestiona dispositivos, horarios, bloqueos, alertas, reportes y autenticación mediante servicios REST e íntegra mediante la API de Groq el modelo llama-3.3-70b-versatile para buscar dominios de forma automática y clasificar sitios web en función de las palabras claves.

Capa Backend (Raspberry Pi – Núcleo del Sistema)

Dentro de esta capa se llega a integrar el almacenamiento tanto de los datos como también de los servicios de red necesarios para el funcionamiento del sistema. MariaDB almacena usuarios, dispositivos, horarios, alertas y reportes, mientras que Pi-hole utiliza SQLite para gestionar los dominios bloqueados y permitidos. Además, incorpora Pi-hole, iptables, Suricata, hostapd y dnsmasq para aplicar las políticas de control parental, filtrar el tráfico, detectar amenazas, administrar la red WiFi y actualizar los bloqueos DNS en tiempo real.

Capa Router (Acceso hacia Internet)

Finalmente, el Router constituye la última etapa de la arquitectura. La función específica que esta infraestructura tiene es redirigir el tráfico procesado y controlado por la Raspberry Pi hacia Internet. En otras palabras, el router no aplica ninguna regla de filtrado, ya que en este caso simplemente se establece como el enlace físico y lógico entre la red local y la red externa, de manera que todo el tráfico de la red local que desee salir a Internet necesite inevitablemente pasar por la infraestructura del control parental. Con ello aseguramos que el tráfico de los usuarios pase obligatoriamente por el control parental antes de salir a Internet.

3.5.2 Diagrama de Conexión del Sistema

El sistema está consolidado por algunos factores que lo ayudan a funcionar de manera autónoma siempre y cuando se respete el diagrama de conexión. A continuación, se presenta el diagrama que ilustra la topología física y de red implementada para el prototipo de control parental, donde la Raspberry Pi 5 actúa como el nodo central del sistema.

A nivel de hardware, la SBC (Single Board Computer) es alimentada por una fuente de tensión continua de tan solo 5.1 voltios y es conectada físicamente al router del proveedor de servicios de internet mediante un cable Ethernet (RJ45). De esta manera, la conexión se establece entre el puerto Gigabit Ethernet de la Raspberry Pi y uno de los puertos LAN que tiene el router y,

con ello, la SBC actúa como un intermediario que encamina el tráfico hacia la red externa. En esta configuración, el router del ISP es exclusivamente el puente físico y lógico hacia el internet, mientras que la SBC se encarga de la parte del procesamiento y de las reglas de filtrado.

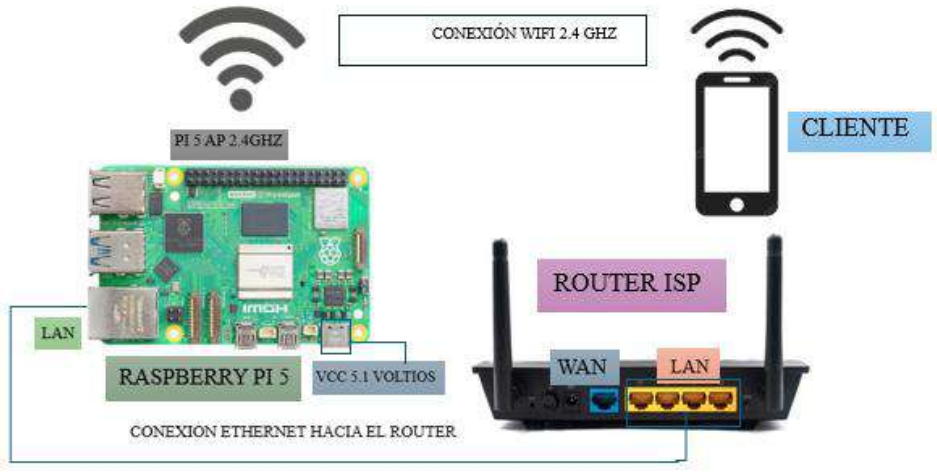
Por otra parte, para habilitar la conectividad de los usuarios, la Raspberry Pi 5 hace uso de su propia interfaz de red inalámbrica para comportarse como un Punto de Acceso (AP) emitiendo una señal Wi-Fi bajo la banda de 2.4 GHz. Los dispositivos clientes (por ejemplo, el teléfono móvil) se conectan directamente a la red Wi-Fi. Además, se añade una interfaz virtual tailscale0 que habilita una VPN de tipo malla gracias al servicio Tailscale. Esta interfaz permite a la aplicación móvil del padre de familia conectarse con la API del sistema independientemente de si se encuentra conectada o no a la red WiFi emitida por la Raspberry Pi, por ejemplo, cuando no se encuentra en el hogar y utiliza datos móviles. La aplicación implementa una lógica de detección automática de servidor que prueba primero la conexión directa por WiFi y, en caso de no obtener respuesta, recurre a la dirección de Tailscale como ruta alterna, todo esto para cuando este en uso la aplicación por los tutores.

Esta arquitectura cumple con el requerimiento técnico de disponer de dos interfaces de red separadas una inalámbrica para la red local de clientes y una cableada para la salida a internet

De esta manera, se garantiza que todo el tráfico generado por los dispositivos conectados pase de forma obligatoria por los servicios del backend alojados en la Raspberry Pi incluyendo el servidor DNS, sistema IDS y el módulo de análisis heurístico antes de ser autorizado para salir hacia internet.

Figura 6

Diagrama de conexión del proyecto

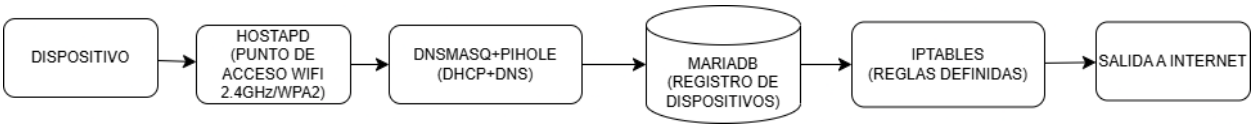


3.5.3 Módulo de Red

Este primer módulo define como un dispositivo que va a ser monitoreado obtiene acceso a la red doméstica y sale hacia Internet. Su diseño se describe mediante un diagrama de bloques y un diagrama de flujo.

Figura 7

Diagrama de bloques del módulo de red



El dispositivo que va ser monitoreado ya sea para pruebas o directamente del cliente, como un teléfono móvil, una computadora portátil o una tablet, que intenta conectarse a la red wifi creada por el sistema. Para permitir esta conexión, se realiza un punto de acceso haciendo uso de la

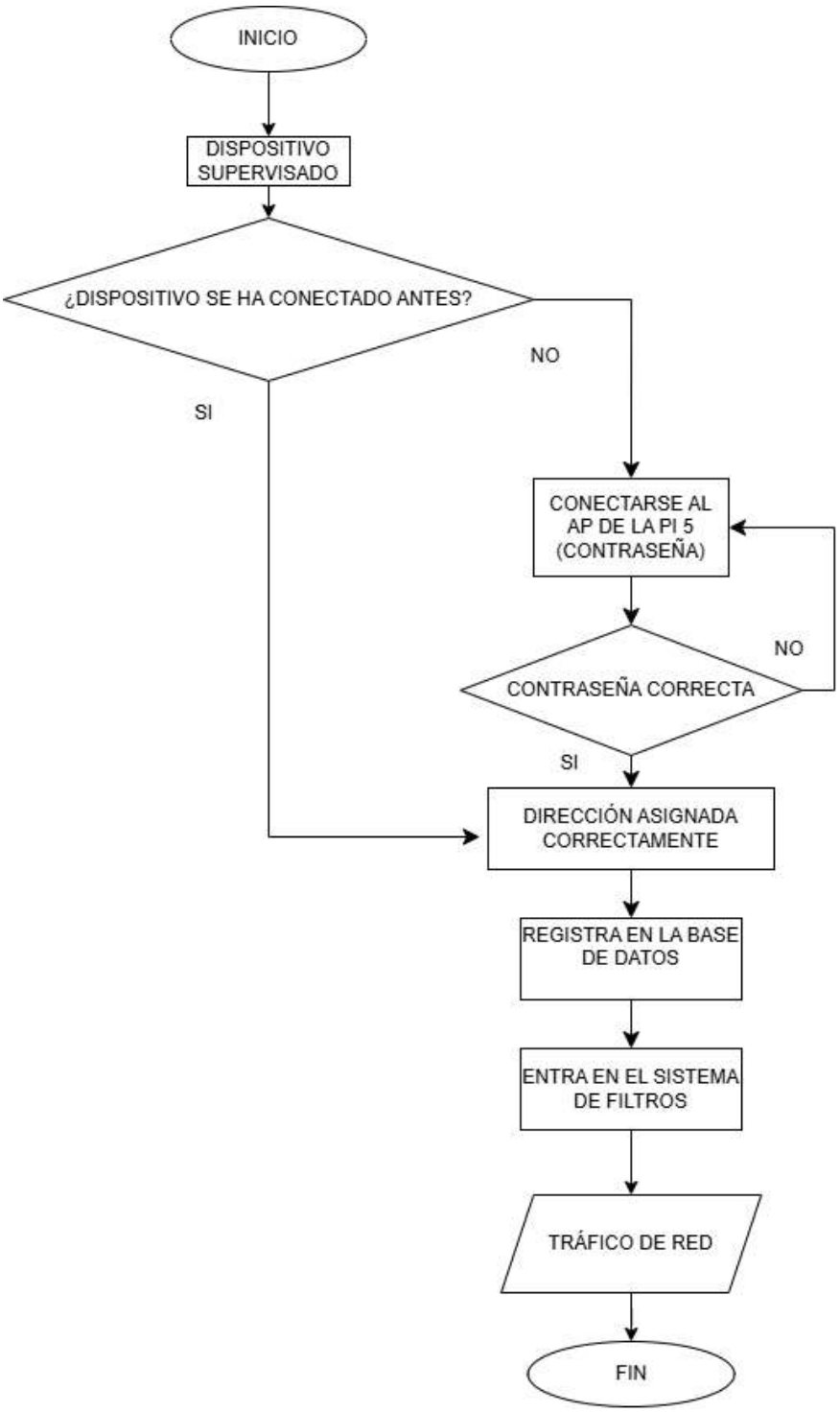
interfaz wlan0 de la Raspberry Pi, haciendo posible que los dispositivos detecten la red y se conecten de forma segura mediante protocolos de seguridad inalámbricas. Una vez establecida la conexión, dnsmasq se usará como servicio de DHCP y Pi-hole como servidor DNS, es decir esto dará la IP a cada dispositivo. Cabe recalcar que Pi-hole además de ser servidor DNS, también bloqueará la publicidad de las páginas web.

Por consiguiente, Las reglas definidas, se encargarán de dirigir el tráfico de datos desde la Raspberry Pi hacia la interfaz Ethernet, realizando la traducción de direcciones de red NAT, con esto el dispositivo puede acceder a Internet.

La comunicación continúa a través del router del proveedor de servicios de Internet (ISP), que representa el punto de salida hacia la red externa. Finalmente, MariaDB almacena la información de cada dispositivo, con el fin de llevar un registro al momento de usar la aplicación móvil.

Figura 8

Diagrama de flujo del módulo de Red (conexión y autenticación)



3.5.4 Módulo de Gestión de Servicios

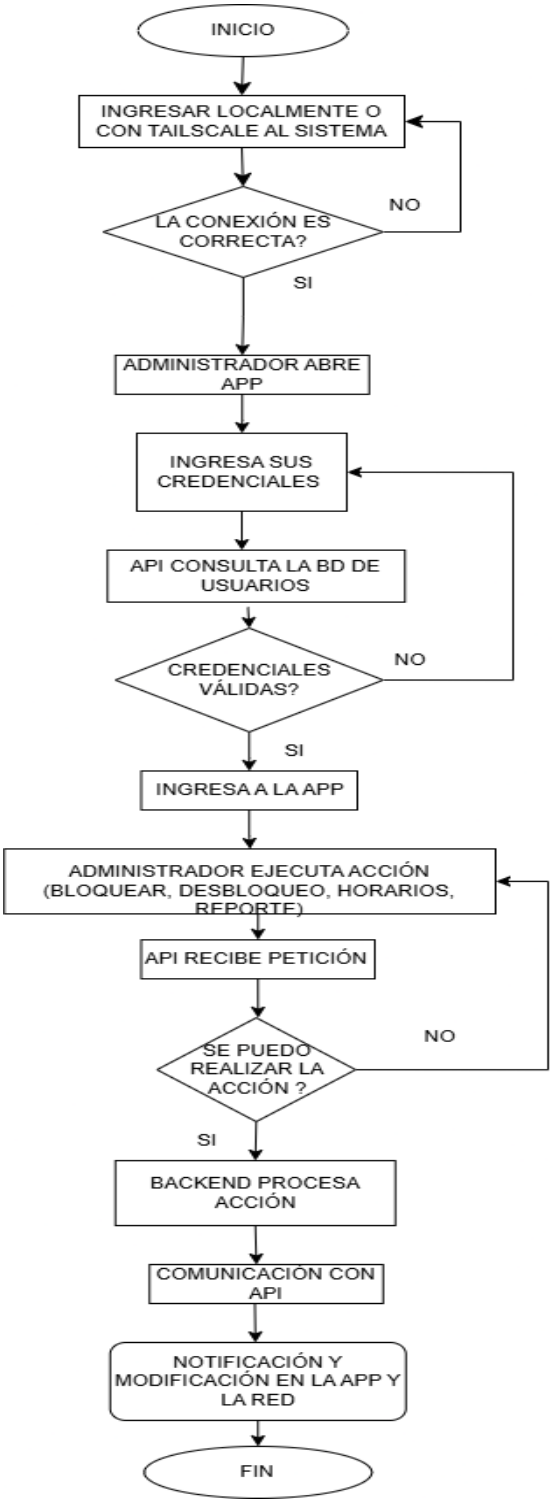
Este módulo define la lógica mediante la cual se generan y aplican las reglas de bloqueo y las políticas de acceso que administran el tráfico de red, sirviendo de base principal para los servicios Pi-hole, iptables, Suricata y MariaDB.

Para llevar esto acabo el administrador debe acceder a la aplicación mediante un proceso de autenticación, ingresando su nombre de usuario y contraseña. La API verifica las credenciales utilitzando el backend y, si son correctas, permite el acceso al Dashboard, si no, se muestra un mensaje de error en el que invita al usuario a volver a intentarlo.

Una vez autenticado, el usuario puede realizar acciones como bloquear o desbloquear sitios web, configurar horarios de acceso a Internet o consultar reportes diarios. Cada solicitud es enviada mediante la API, la cual actúa como un intermediario entre la aplicación móvil y el backend, validando la información y ejecutando la correspondiente operación. Si el proceso se completa correctamente, la acción se aplica o de lo contrario, se notifica el error al usuario y debe volver a intentar.

Figura 9

Diagrama de flujo de módulo de Gestión de Servicios

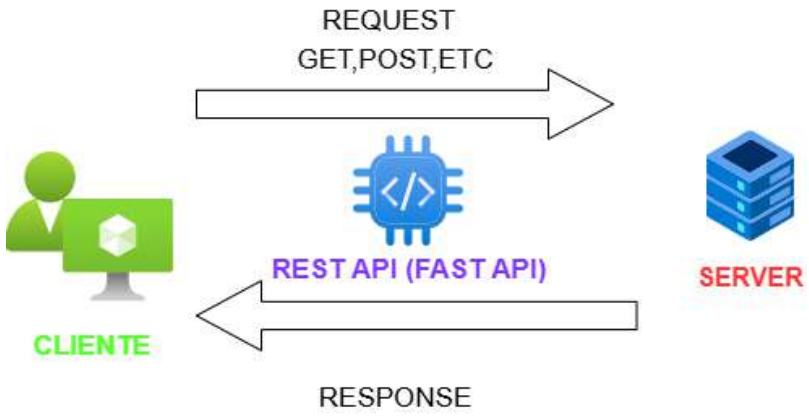


3.5.5 Módulo de Conexión Lógica

Para establecer la comunicación entre el backend y el frontend, fue necesario definir un mecanismo que permitiera el intercambio de información entre ambos componentes. Para ello, se implementó una API REST, la cual actúa como intermediaria entre la aplicación móvil y el servidor, permitiendo el envío y recepción de solicitudes relacionadas con la configuración, consulta y administración del sistema.

Figura 10

Interacción de API REST



En el siguiente diagrama de flujo se detalla el proceso de interacción entre el administrador que usa la aplicación móvil que vendría a ser el frontend y el sistema dentro de la Pi 5 que es el backend del sistema.

El proceso inicia cuando el usuario abre la aplicación móvil e ingresa sus credenciales. La información se envía a la API REST, desarrollada con FastAPI, que verifica los datos en la base de datos MariaDB. En caso de que la validación no sea correcta, se deniega el acceso y se

muestra un mensaje de error en la aplicación, mientras que en caso de que la validación sea correcta, se accederá al panel principal.

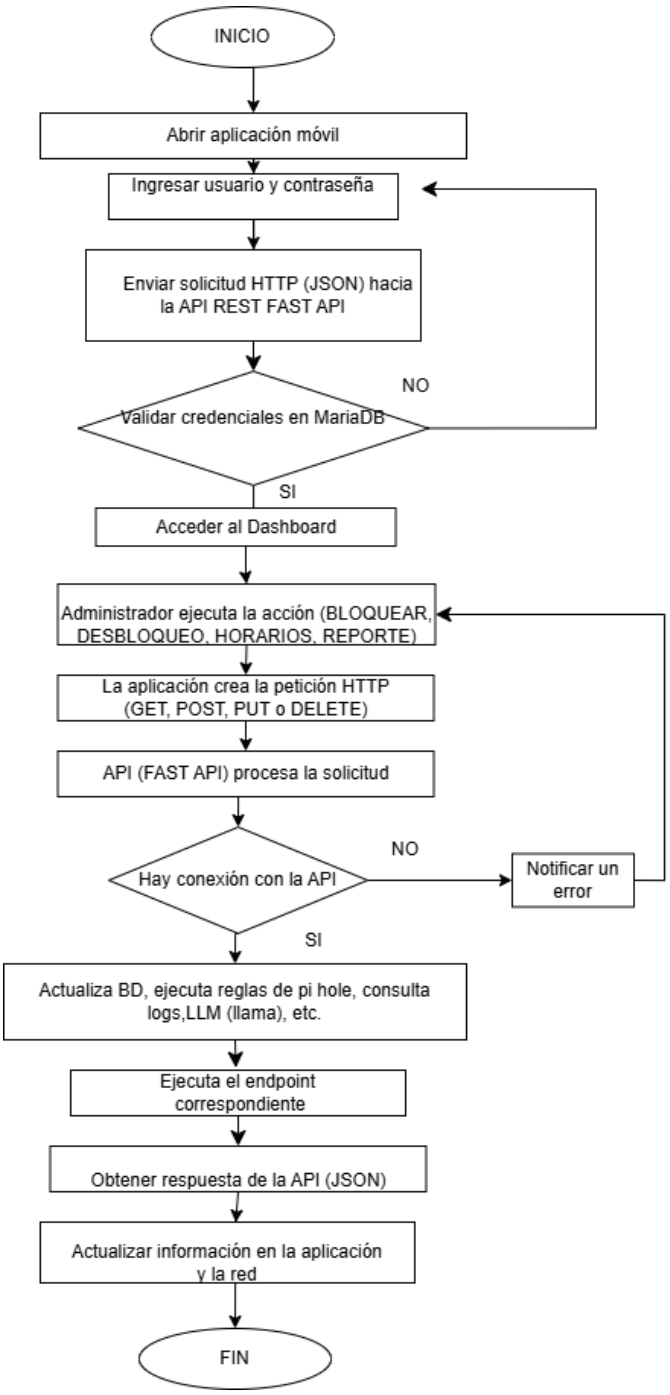
Desde nuestro dashboard, se podrá realizar una serie de acciones, como el hecho de bloquear o desbloquear un sitio web, consultar los dispositivos conectados, configurar el horario de navegación, monitorizar la actividad de los dispositivos o visualizar reportes.

Cuando solicitamos bloquear un sitio web, la aplicación manda la palabra clave a la API, la cual hace uso del modelo de lenguaje de Groq para obtener la lista de dominios y subdominios que estén relacionados. Con esta información, nuestra Raspberry Pi 5 actualiza las listas de bloqueos de Pi-hole. Para todas las demás funciones, la aplicación mandará la configuración en formato JSON y la API procesa la solicitud actualizando las tablas de la base de datos y realizando todos los cambios necesarios en el sistema.

La API al final manda una respuesta a la aplicación para especificar si se ha realizado correctamente la operación, en el caso de que sean reportes, la API consultará la información almacenada en el backend y la devuelve a la aplicación para poder visualizar las estadísticas y datos de monitorización. Una vez completada la acción, el usuario regresa al dashboard para continuar administrando el sistema.

Figura 11

Diagrama general de flujo de conexión lógica (API)



3.5.5.1 Configuración de la API

La API integra procesos para diferentes servicios y así realizar el monitoreo continuo, la comunicación con las bases de datos MariaDB y SQLite, IDS, Ipatables, etc. La API se organizó en 7 módulos funcionales con un total de 63 endpoints.

Tabla 15

Estructura y distribución de endpoints

Módulo	Endpoints principales	Función
Autenticación	/login	Validación de credenciales del administrador
Dispositivos	/dispositivos-activos /dispositivos-bloqueados /dispositivos-apodo	Gestión y control de dispositivos
Horarios	/horarios/crear /horarios/eliminar /horarios	Control de acceso por tiempo
Bloqueo/Desbloqueo	/listar-reglas /agregar-regla /eliminar-regla	Bloqueo y desbloqueo de sitios web con ayuda de la IA
Reportes/IA	/generar-reporte /reporte-PDF/ reporte-IA/historial/Score	Generación de informes y análisis
Alertas	/alertas /notificaciones /resumen	Sistema de notificaciones y alertas
ChildGuard	/registrar /bloqueo-app/reportar /actividad/configuración/estado	Control del celular del hijo

Cada módulo cumple una función específica, aunque todos trabajan de forma integrada para garantizar el correcto funcionamiento del sistema. La información generada por los distintos módulos converge en el módulo de Reportes, el cual recopila los datos almacenados por la API,

los procesa y genera automáticamente el informe en formato PDF con las estadísticas y eventos registrados por el sistema.

3.5.6 Módulo de monitorea e interfaz de usuarios

El sistema se desarrolló mediante dos aplicaciones independientes. Una está orientada al administrador, quien puede gestionar la red y los dispositivos conectados. La otra está destinada al usuario que será monitoreado, permitiendo únicamente las funciones definidas para su perfil, mientras el sistema supervisa de forma continua su actividad en la red.

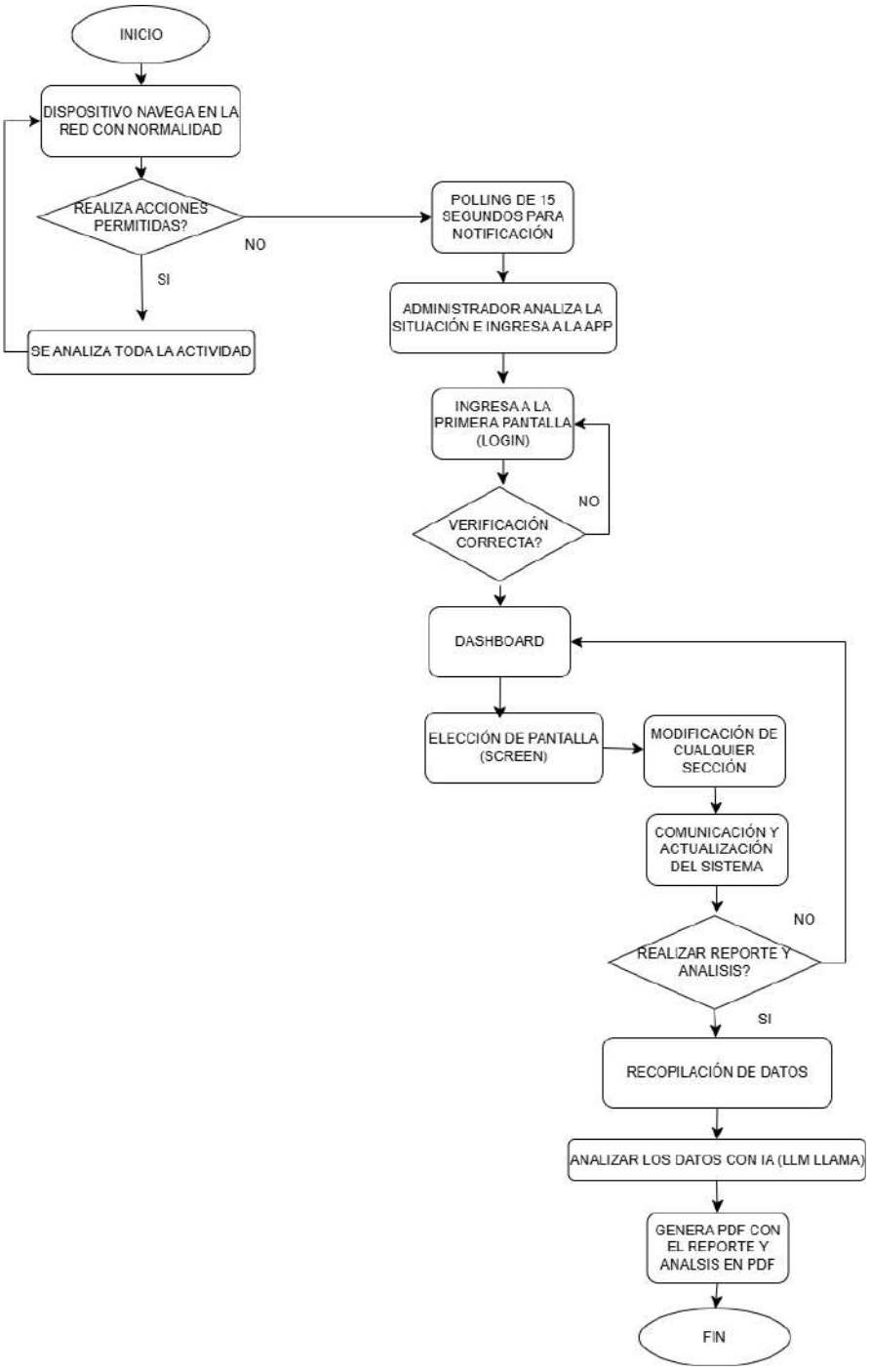
3.5.6.1 Aplicación del administrador

La aplicación del administrador incorpora un sistema de notificaciones que alerta cuando se detecta una actividad que requiere atención. El administrador revisar la notificación o alerta e ingresa a la aplicación móvil, realizando el respectivo registro.

Tras haber accedido al panel principal, el administrador dispone de distintas opciones, entre las cuales se incluyen, la de bloquear/desbloquear cualquier sitio web, consultar los dispositivos conectados, configurar un horario de acceso, ver la actividad de la red e incluso generar reportes. Cada acción disparada se envía a la API la cual ejecuta la acción y devuelve la respuesta que queda reflejada en la aplicación, entre tanto el sistema consulta cada 15 segundos si se han generado nuevas alertas y es capaz de mostrarlas mediante una correspondiente notificación. Después de cada operación, la información se actualiza automáticamente y el usuario regresa al panel principal para continuar administrando la red. Cabe recalcar que toda actividad se registra en la base de datos para realizar un análisis del comportamiento de cada dispositivo y realizar los reportes correspondientes.

Figura 12

Diagrama de flujo del módulo de la interfaz del adminstrador



La interfaz fue diseñada bajo un diseño amigable para el administrador basada en componentes, donde cada módulo funcional se implementa como una pantalla independiente, facilitando la reutilización del código, el mantenimiento y la escalabilidad del sistema.

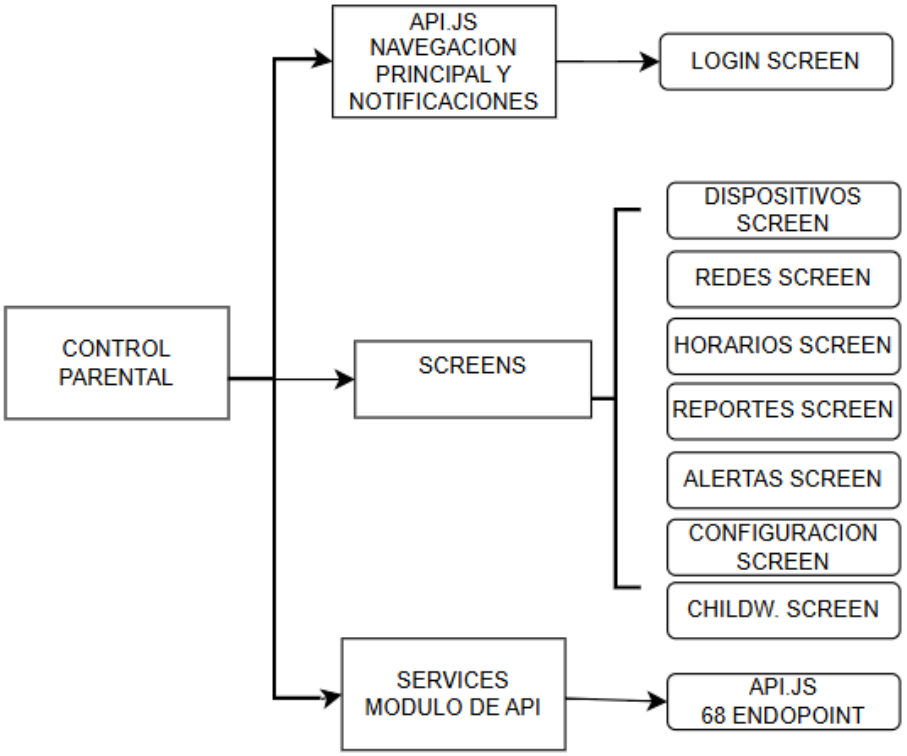
La navegación entre pantallas se realiza mediante React Navigation, proporcionando una experiencia de usuario fluida e intuitiva.

La comunicación con el servidor se implementó mediante peticiones HTTP utilizando la librería Axios, la cual consume los servicios REST expuestos por la API desarrollada en FastAPI.

Cada solicitud intercambia información en formato JSON para realizar procesos de autenticación, administración de dispositivos, gestión de horarios, control de redes sociales, consulta de alertas y generación de reportes inteligentes.

Figura 13

Estructura de la aplicación



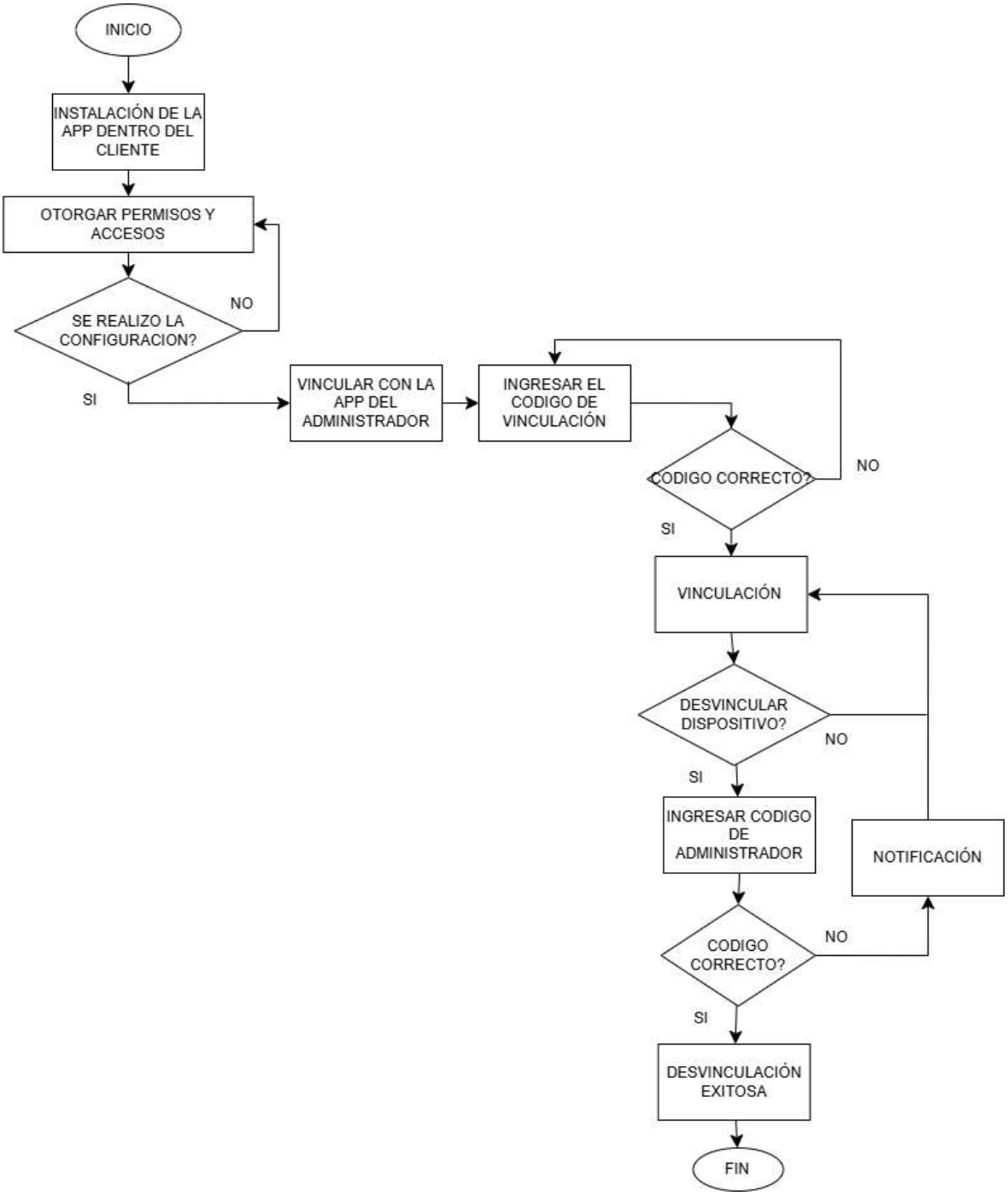
3.5.6.2 Aplicación del cliente (cliente)

El flujo de la aplicación se diseñó para garantizar una configuración segura desde el primer uso. Primeramente, el usuario debe conceder los permisos necesarios para que la aplicación funcione correctamente. Luego, el dispositivo se vincula con la aplicación del administrador mediante un código de acceso.

Una vez completada la vinculación, el dispositivo permanece bajo supervisión hasta que el administrador autorice su desvinculación. Para evitar accesos no autorizados, cualquier intento de desvincular el dispositivo requiere la validación del código del administrador y, si el código es incorrecto, el sistema envía una notificación al administrador. Es una aplicación minimalista y simple solo para monitorear el dispositivo en tiempo real.

Figura 14

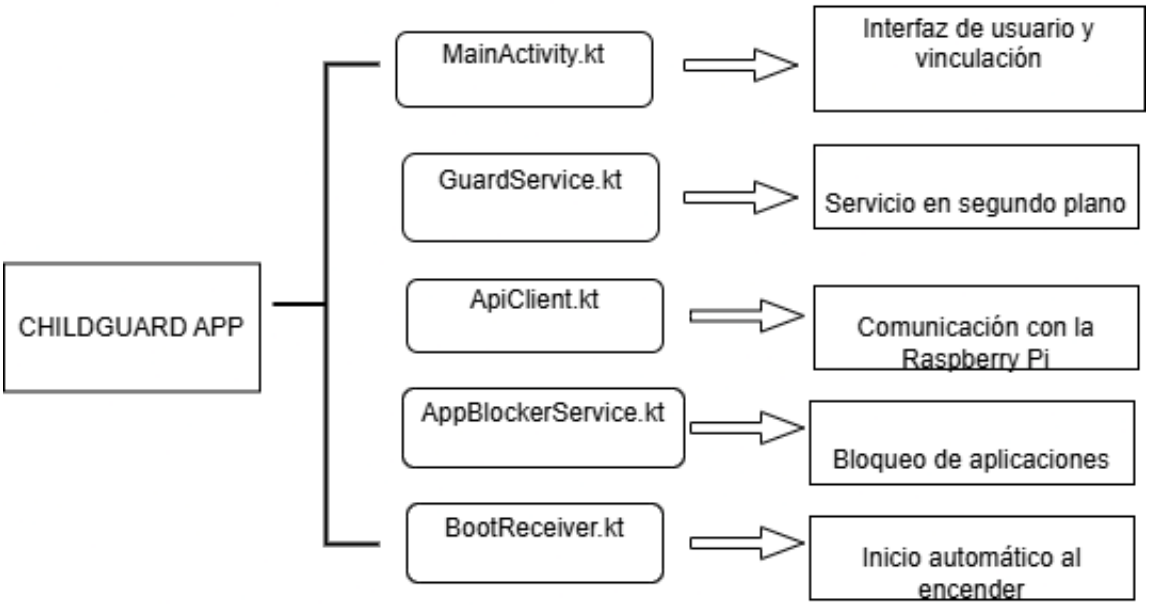
Diagrama de la aplicación a monitorear (Childguard)



La aplicación se estructuró en cinco archivos principales. Cada uno reúne las funciones correspondientes a un módulo específico, lo que facilita la organización al momento de realizar la recopilación de datos del dispositivo.

Figura 15

Estructura de la aplicación Childguard



3.5.7 Módulo de análisis heurístico

En el siguiente módulo se define cómo el sistema evalúa el riesgo asociado al comportamiento de navegación de cada dispositivo conectado a la red y que está siendo monitoreado por el sistema de control parental.

El proceso inicia cuando los dispositivos conectados generan tráfico de red, el cual es supervisado por cada uno de los servicios. Por ejemplo, Pi-hole inspecciona las consultas DNS y aplica listas de bloqueo para identificar dominios restringidos, clasificando además cada dominio detectado según su nivel de riesgo mediante reglas predefinidas o, en caso de no estar previamente catalogado, mediante consulta a un modelo de inteligencia artificial, además, automáticamente

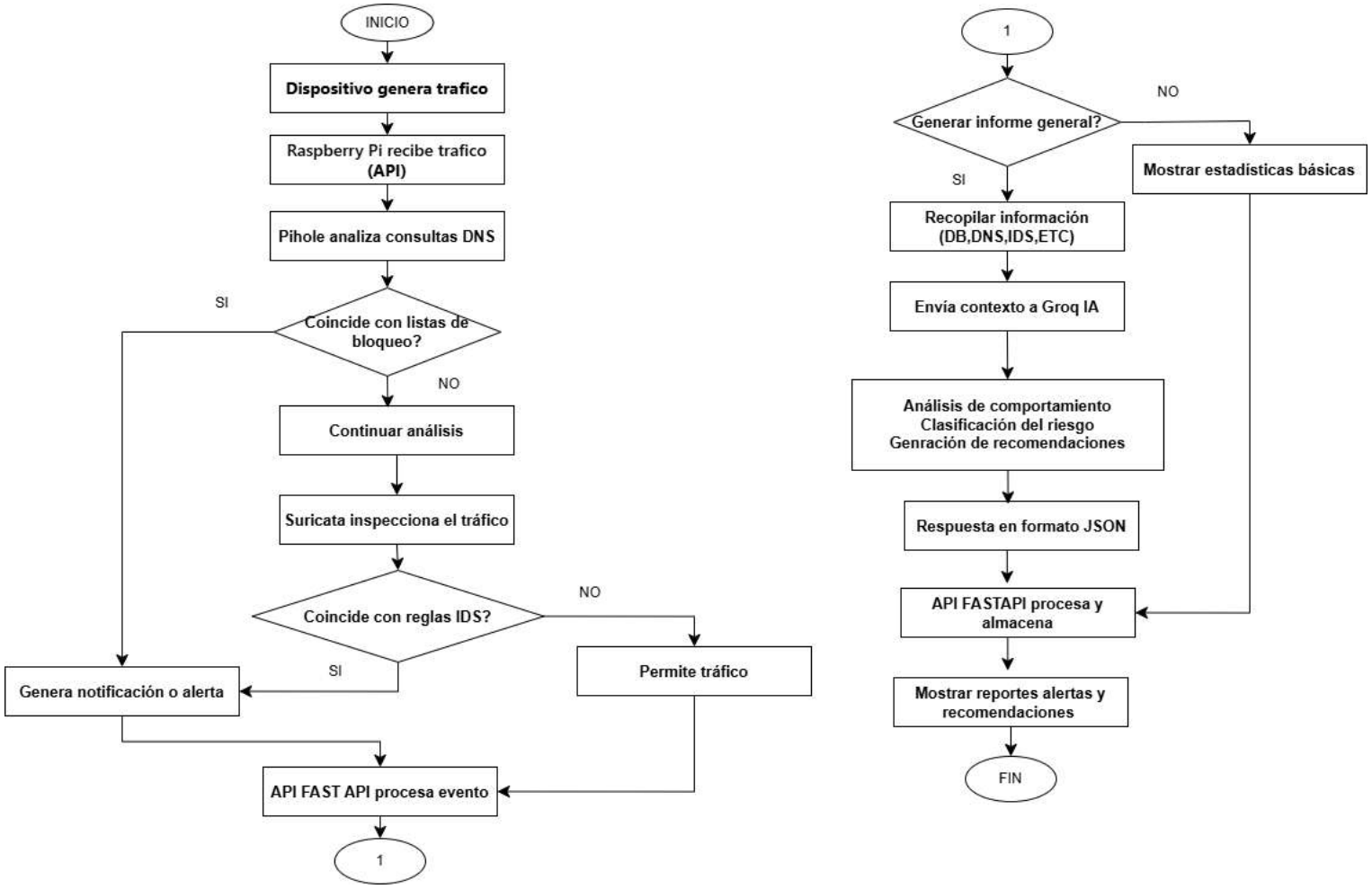
estará bloqueando la publicidad. De forma continua, Suricata analiza el tráfico mediante reglas IDS para detectar posibles amenazas o comportamientos anómalos.

Los eventos o actividades obtenidos son enviados a la API REST desarrollada en FastAPI, donde se consolidan y almacenan en MariaDB, gestionando además las sesiones de navegación de cada dispositivo para evitar la generación repetitiva de notificaciones ante un mismo evento. Posteriormente, cuando el tutor solicita un reporte inteligente o cuando el sistema requiere un análisis, la API recopila la información registrada y la envía a una IA en este caso un LLM (llama-3.3-70b-versatile), desarrollado por Meta AI y desplegado a través de la plataforma de inferencia Groq,

La inteligencia artificial evalúa el historial de navegación, los dominios bloqueados, las amenazas detectadas y los patrones de uso, generando una interpretación del comportamiento, un nivel de riesgo y recomendaciones orientadas a fortalecer el control parental. Finalmente, el resultado es almacenado en la base de datos con un tamaño decente para que no afecte a la BD, si es necesario realizar el reporte, tomara esos datos y se entrega en formato PDF.

Figura 16

Diagrama del análisis heurístico



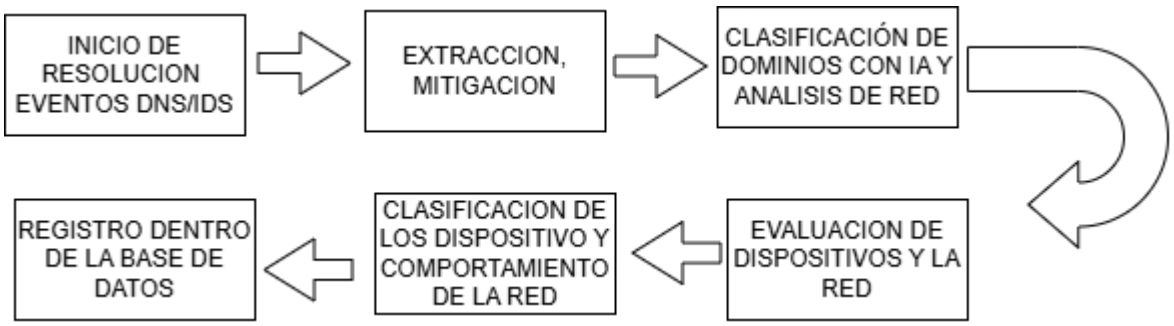
Una vez que el administrador y el dispositivo monitoreado han interactuado con la aplicación y antes de generar el informe en formato PDF, se realiza un análisis del estado de cada dispositivo para determinar su comportamiento dentro de la red. Este proceso se desarrolla por fases, lo que permite evaluar la información recopilada de manera ordenada y obtener una visión clara del uso y la actividad de cada dispositivo.

3.5.7.1 Enfoque metodológico y analítico

Este algoritmo describe la extracción de eventos de red, la clasificación inteligente de dominios, el cálculo matemático del score de riesgo y la persistencia optimizada de datos en el servidor.

Figura 17

Diagrama de bloques sobre el método de análisis heurístico



Fase 1: Extracción, Normalización y Mitigación de Ruido

El sistema se comunica con Pi-hole y Suricata de dos formas. Primero, revisa los registros recientes para mostrar la actividad en el panel de la aplicación. En segundo lugar, monitorea en todo momento los nuevos eventos con el fin de identificar de forma inmediata cuando se trata de acceder a una página bloqueada. A su vez, automáticamente ignora todos los bloqueos por publicidad o servicios de fondo, evitando así enviar alertas innecesarias a los padres.

Fase 2: Clasificación de Dominios y Control de Sesión

El sistema primero busca las listas de palabras clave y las listas de dominios que ya están clasificados. Si no encuentra ninguno de los anteriores, envía consultas a la LLM (llama) de Groq para determinar la categoría a la que pertenece un dominio no conocido. Además, para evitar enviar muchas alertas por el mismo sitio, agrupa los intentos realizados en un periodo de cinco minutos

y solo notifica cuando es necesario. De la misma manera Suricata realiza un análisis de la red si hay anomalías o un tráfico inusual mediante sus reglas preestablecidas.

Fase 3: Formalización del Algoritmo de Puntuación (Score)

El sistema calcula un puntaje de riesgo para cada dispositivo tomando en cuenta la actividad registrada durante las últimas 24 horas. Para ello analiza cinco aspectos, que son intentos de acceso a contenido para adultos, sitios peligrosos, plataformas de apuestas, redes sociales bloqueadas y navegación fuera del horario permitido.

Cada tipo de actividad suma una determinada cantidad de puntos según la frecuencia con la que ocurre. En algunos casos se establece un pequeño margen de tolerancia para evitar que anuncios o procesos automáticos generen un puntaje innecesario. El resultado final se limita a un máximo de 100 puntos, donde un valor más alto indica un mayor nivel de riesgo y permite a los padres identificar de forma rápida los dispositivos que requieren mayor supervisión.

Fase 4: Clasificación de los dispositivos

Una vez calculado el puntaje de riesgo, el sistema clasifica cada dispositivo en tres niveles bajo, medio o alto. Cada nivel se representa con un color en la aplicación verde para riesgo bajo, naranja para riesgo medio y rojo para riesgo alto, facilitando a los administradores identificar rápidamente la situación de cada dispositivo. A su vez Suricata realiza una pequeña recopilación de información y determina si es algo peligroso dentro de la red. Finalmente, la información se guarda en la base de datos. Si ya existe un registro del mismo dispositivo para ese día, este se actualiza en lugar de crear uno nuevo, evitando duplicados y mejorando el rendimiento de la Raspberry Pi.

3.6 Implementación

En esta sección se presenta la implementación de cada uno de los módulos definidos en el diseño del sistema. Se describe de forma secuencial el proceso de instalación y configuración de los componentes de software en la SBC y en la aplicación móvil, es decir, se muestra el desarrollo del backend y del frontend a partir de la arquitectura propuesta. Por otra parte, se describe el montaje del hardware y se presentan las conexiones físicas necesarias para poner en funcionamiento por completo el sistema. Por módulos, se desarrolla individualmente la estructura del mismo, la configuración de cada uno de los módulos y todos los elementos que permiten la integración y el funcionamiento del mismo en el prototipo.

3.6.1 Implementación de Software

En cuanto a la implementación del software se puede mencionar que se llevó a cabo sobre el sistema operativo Raspberry Pi OS, logrando tener presente una instalación en una tarjeta microSD de 128 GB. El objetivo fue transformar la Raspberry Pi 5 en un dispositivo multifuncional que actúa simultáneamente como un punto de acceso, con servicios como servidor DHCP, servidor DNS con bloqueo de contenido, IDS, reglas iptables, API REST, VPN y aplicación móvil.

3.6.2 Módulo de red

De acuerdo con lo mencionado dentro del módulo de red en el diseño del sistema, en esta sección se implementó la infraestructura de red que sustenta el sistema de control parental, tomando como núcleo operativo la Raspberry Pi 5. El objetivo principal fue transformar este dispositivo en un punto de acceso inalámbrico capaz de gestionar tráfico, asignar direcciones IP, resolver consultas DNS, aplicar reglas, hacer uso de VPN y analizar la red.

3.6.2.1 Configuración de la plataforma SBC como Punto de Acceso

Para que la Raspberry Pi 5 opere como el nodo central de la red doméstica y gestione el tráfico de los dispositivos clientes, fue necesario transformar su interfaz inalámbrica en un punto de acceso. Este procedimiento técnico se estructuró en tres fases fundamentales que son:

- Configuración de direccionamiento estático
- Emisión de la red inalámbrica
- Asignación dinámica de direcciones.

3.6.2.1.1 Fase 1: Asignación de direccionamiento estático

El primer paso consistió en modificar la interfaz de red inalámbrica wlan0 en este caso de los procesos de asignación automática del sistema operativo. Mediante la modificación del dhcpcd, se estableció una dirección IPv4 estática privada 192.168.10.1 con máscara de subred /24.

interface wlan0

static ip_address=192.168.10.1/24

nohook wpa_supplicant

Al desactivar wpa_supplicant, la dirección IP configurada se convierte en la puerta de enlace de la red del control parental. Esto permite que la Raspberry Pi dirija el tráfico de los dispositivos conectados y aplique las funciones de control definidas en el sistema. Es decir, pasa de ser un cliente a ser un servidor de internet.

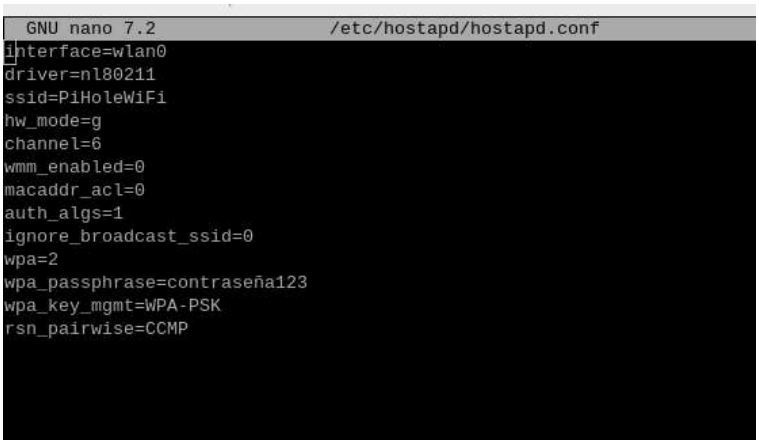
3.6.2.1.2 Fase 2: Despliegue del servicio de emisión Wi-Fi

Para la gestión del hardware de red y la transmisión de la señal de forma inalámbrica se utilizó el servicio hostapd, cuya parametrización se configuró para el estándar IEEE 802.11g y la

frecuencia de 2.4 GHz, seleccionado además el canal 6 de transmisión para evitar interferencias de radioformas que pueda haber en el entorno doméstico. En el campo de la seguridad, el enlace inalámbrico se protegió mediante el protocolo WPA2. Esta configuración asegura la confidencialidad e integridad de la comunicación en la capa de enlace de datos.

Figura 18

Hostpot de Pi 5 revisar el canal

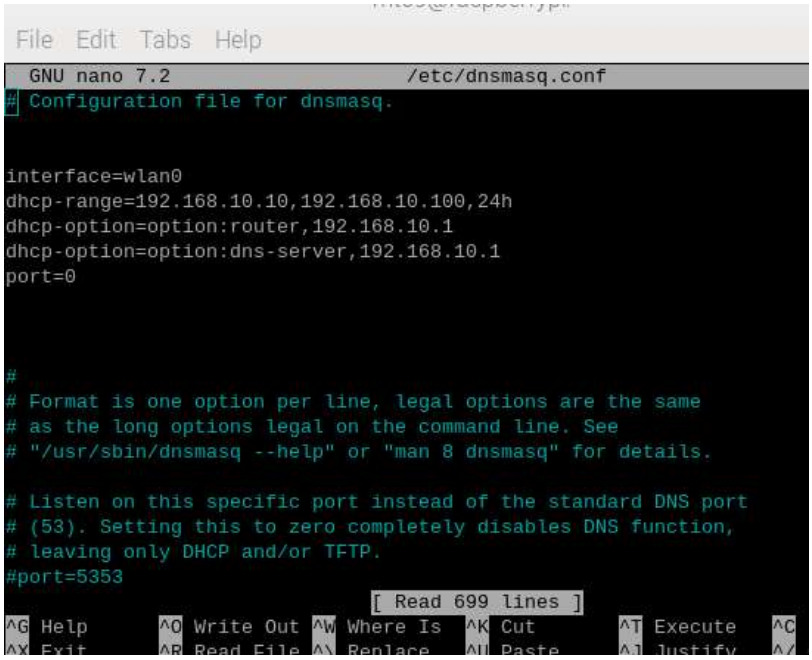


3.6.2.1.3 Fase 3: Implementación de los servicios DHCP

Para el proceso de conexión y aplicación de IP de manera automática se implementaron dos servicios. Por un lado, se instaló dnsmasq, el cual fue configurado para cumplir como servidor DHCP, estableciendo un pool de direcciones IP dinámicas desde la 192.168.10.10 hasta la 192.168.10.100.

Figura 19

Configuración de DHCP con dnsmasq



Para la administración del servicio DNS se utilizó Pi-hole en lugar de dnsmasq. Aunque dnsmasq es una herramienta ligera y adecuada para funciones básicas de DNS y DHCP, sus opciones de administración y filtrado son limitadas para un sistema de control parental. Por esta razón, se optó por Pi hole, ya que ofrece una gestión más completa de listas de bloqueo, un mejor monitoreo de las consultas DNS y una interfaz de administración que facilita la configuración y el control del sistema.

3.6.3 Módulo de gestión de servicios

Este módulo reúne los servicios principales encargados de controlar el tráfico de datos, almacenar la información y aplicar las reglas de control parental. Para su implementación se configuró el filtro DNS, IDS, iptables y la base de datos, organizando cada elemento para que funcione de manera integrada dentro del sistema.

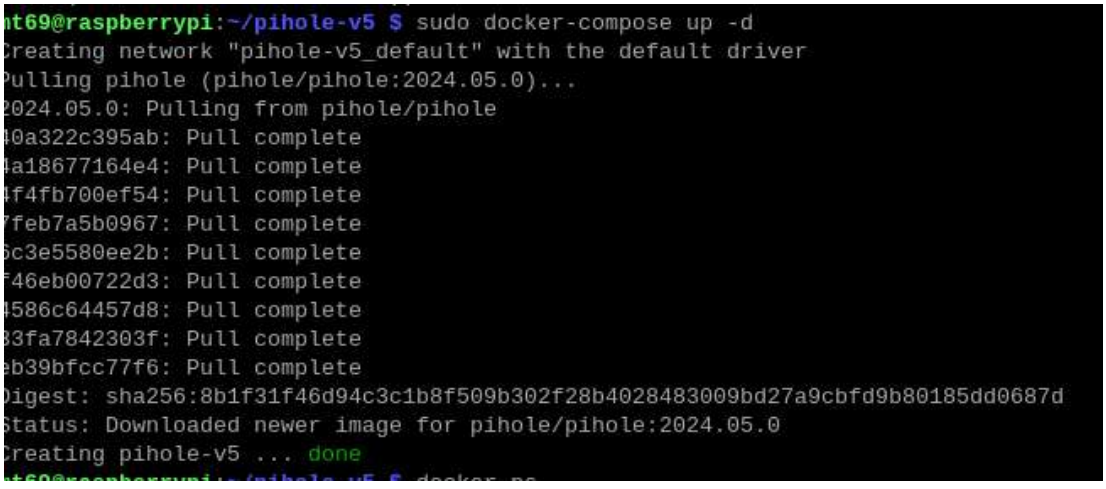
3.6.3.1 Despliegue del Sistema de Filtrado DNS (Pi-hole)

Para la resolución de nombres de dominio y el bloqueo automático de publicidad o sitios maliciosos, se implementó Pi hole. Con el propósito de optimizar los recursos de la Raspberry Pi y aislar las dependencias del software, se llevó a cabo la instalación a través de Docker, que es una tecnología de virtualización muy utilizada en contenedores. Para ello, en primer lugar, se instaló Docker en la Raspberry Pi 5 para ejecutar el servicio a través de un contenedor. A continuación, se crean los directorios para almacenar la configuración y los datos de Pi-hole v5, lse lo hace en Docker para evitar actualizaciones automáticas y problemas de compatibilidad en el futuro. Aunque durante las pruebas el contenedor se inició manualmente, en la implementación final este proceso quedó automatizado, de modo que tanto Docker como Pi-hole se inician automáticamente cada vez que la Raspberry Pi se enciende.

El repositorio contiene todas las configuraciones necesarias para la implementación y funcionamiento del sistema: https://github.com/bryanarotingo-prog/Control-Parental-MT.git

Figura 1

Primer despliegue del contenedor de Pi-hole



El contenedor fue desplegado utilizando el modo de red host, lo que permitió que el servicio se vinculara directamente a las interfaces físicas del sistema, operando sobre los puertos 53, 80 y 443 sin la sobrecarga de una red puente virtual.

Figura 20

Configuración del docker compose

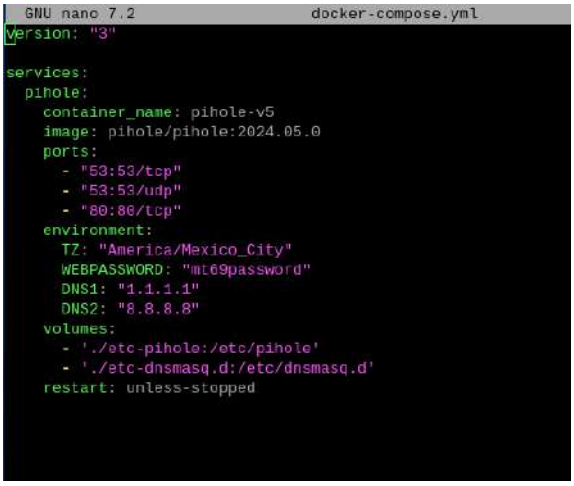


Figura 21

Inicialización del docker

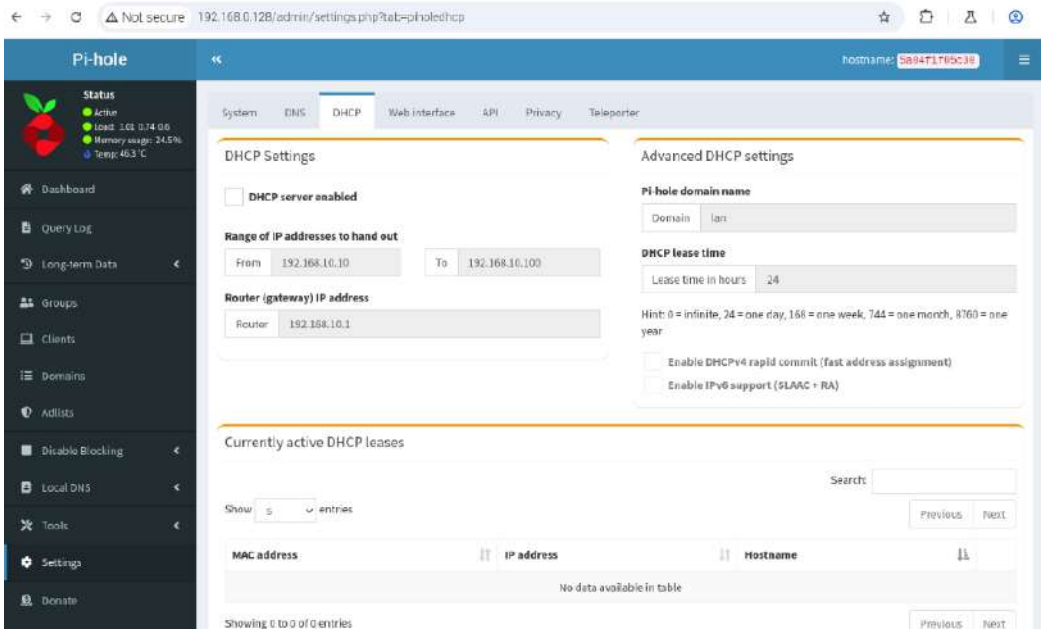


Durante la inicialización, se configuraron volúmenes persistentes para salvaguardar las listas de bloqueo y los archivos de configuración (/etc/pihole y /etc/dnsmasq.d), asegurando que

las reglas establecidas no se perdieran ante reinicios del sistema. A continuación, se presenta la interfaz web de Pi-hole, la cual evidencia que la instalación y configuración del sistema se realizaron correctamente y se encuentra funcionando de manera adecuada.

Figura 2

Dashboard de Pihole V5



En este proyecto se optó por utilizar Pi-hole versión 5, debido a que es una versión estable, ampliamente documentada y respaldada por una gran comunidad de usuarios. Además, ofrece una alta compatibilidad con entornos de virtualización y contenedores, como Docker, lo que facilita su implementación y administración.

Por otro lado, aunque Pi-hole versión 6 corresponde a la versión más reciente, al momento del desarrollo de este trabajo aún se encontraba en un proceso de evolución con cambios y pruebas continuas, lo que podría afectar la estabilidad del sistema. A continuación, se presenta una tabla comparativa entre ambas versiones.

Tabla 16

Comparativa de versiones de Pi hole

Servicio	**Pi-hole v5**	**Pi-hole v6**
Estabilidad	Alta, versión LTS probada.	Sigue en desarrollo activo
API	Básica pero completamente funcional	Sigue en desarrollo con una interfaz más sofisticada.
Compatibilidad Docker	Es apta y compatible	Parcialmente en pruebas
Soporte comunidad	Tiene una extensa y consolidada documentación,	La documentación sigue en construcción.

Para la implementación de Pi-hole se eligió Docker como plataforma de despliegue, ya que permite ejecutar la aplicación de forma separada del sistema operativo principal. Esta característica evita posibles conflictos con otros servicios utilizados en el proyecto, como FastAPI, Suricata y hostapd. Asimismo, el uso de contenedores facilita las tareas de mantenimiento, puesto que, ante una actualización o un inconveniente en el servicio, Pi-hole puede restaurarse o volver a desplegarse rápidamente sin comprometer el funcionamiento del resto de los componentes del sistema.

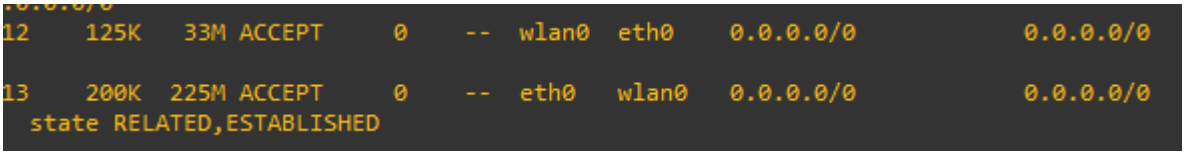
3.6.3.2 Configuración del iptables

La realización de la configuración de las reglas de iptables es una parte esencial para el funcionamiento del sistema, ya que permite el control del flujo del tráfico entre las diferentes interfaces de red. En este supuesto se implantaron dos reglas principales:

- La primera es la que permite que los dispositivos conectados a la red inalámbrica puedan acceder a Internet a través de la interfaz Ethernet.
- La segunda es la que permite el retorno de las respuestas sólo para aquellos que correspondan a conexiones que ya han sido establecidas, evitando que se admitan conexiones externas (no solicitadas) de ningún tipo.

Figura 22

Reglas de Reenvío (FORWARD)



Por otro lado, tenemos la regla NAT, para el tráfico que sale por la interfaz Ethernet. Esta regla se encarga de traducir las direcciones IP privadas de los dispositivos conectados a la red local, a la dirección IP utilizada por la Raspberry Pi para acceder a Internet. Gracias a este proceso, varios dispositivos pueden compartir una misma conexión a Internet sin afectar la comunicación entre ellos y la red externa.

Figura 23

Regla NAT: Enmascaramiento de red



Otro tipo de regla que se implementó fue la de gestión remoto en este caso se hace uso de Tailscale VPN, esto se lo realiza para cuando el administrador no se encuentre en la red local pueda usar la app en cualquier lugar.

ts-input: Acepta tráfico de la VPN Tailscale

ACCEPT -- tailscale0 * 0.0.0.0/0 0.0.0.0/0

ts-forward: Marca y reenvía tráfico VPN

MARK -- tailscale0 * MARK xset 0x40000/0xff0000

Y para hacer uso del Docker también se realiza reglas de aislamiento. Estas reglas bloquean el puerto 853 desde la red Wifi, obligando a que todo pase por el servidor DNS de Pihole.

DOCKER-USER: Bloquea DNS over HTTPS externo

REJECT -- wlan0 * tcp dpt:853

REJECT -- wlan0 * udp dpt:853

Hay otra regla que se usa cuando el administrador realiza acciones definidas, por ejemplo, Cuando el padre desactiva el internet de un dispositivo desde la pantalla de Dispositivos en la aplicación móvil, el backend recibe la solicitud y ejecuta automáticamente la siguiente regla.

iptables -I FORWARD 3 -s 192.168.10.X -j DROP

Esta regla se inserta dinámicamente en la posición 3 de la cadena FORWARD, identificando al dispositivo por su dirección IP. A partir de ese momento, todos los paquetes que provengan de ese dispositivo son descartados silenciosamente, cortando su acceso a internet sin afectar a los demás dispositivos de la red. Cuando el padre restaura el acceso desde la misma pantalla, el backend elimina la regla con el comando.

iptables -D FORWARD -s 192.168.10.X -j DROP

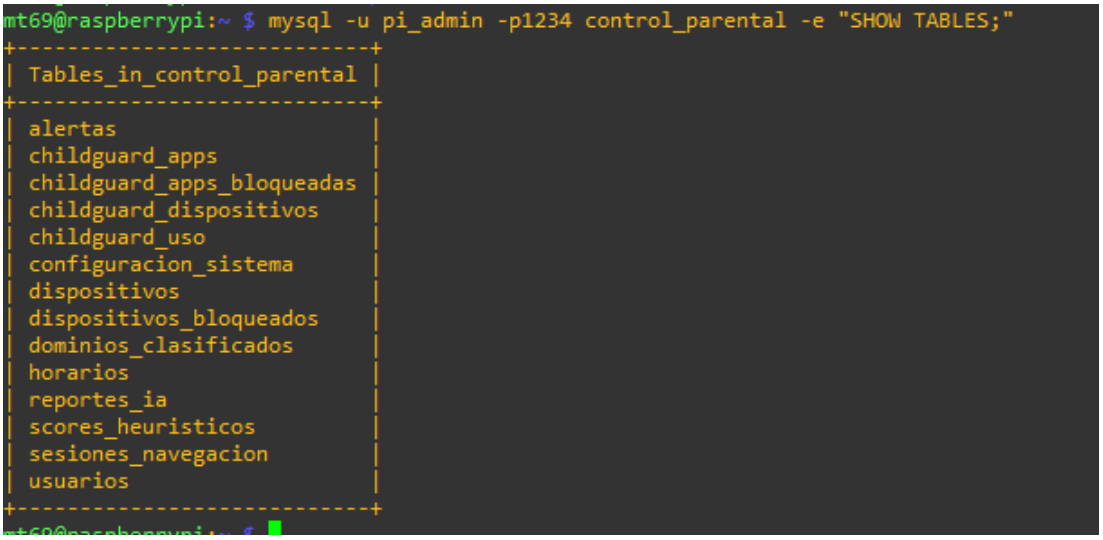
Estas mismas reglas se aplican cuando el administrador realiza la acción de colocar horarios, esto se ejecuta independientemente, es decir, que cada 10 segundos el sistema vuelve a comprobar la hora actual y así determina el horario colocado e inserta las reglas Drop o las elimina.

3.6.3.3 Motor de Base de Datos

Finalmente, para garantizar la integridad y el almacenamiento de la información del sistema, se hizo uso de MariaDB como motor de base de datos. La base de datos opera como el repositorio central del cual el módulo de conexión (API), extraerá la información para validar sesiones y aplicar los cambios solicitados por el administrador.

Figura 24

Base de datos MariaDB



La estructura implementada se basa en la información que el usuario necesite que se despliega en diferentes tablas, cada una destinada a almacenar información específica, como los datos de autenticación, los dispositivos registrados, los horarios de acceso, las alertas y los reportes generados por el sistema.

Esta distribución permite mantener la información ordenada, facilita la gestión de los datos y mejora el procesamiento de las consultas realizadas por la API REST. Asimismo, favorece la integración con la aplicación móvil, permitiendo que toda la información del sistema de control parental se administre de forma ordenada.

3.6.4 Módulo de conexión lógica

Para la inicialización del sistema en la Raspberry Pi 5 se implementó un backend desarrollado en Python 3.11 usando el framework FastAPI que usa el puerto 8000, encargada de gestionar las dependencias, servicios y variables de entorno necesarias para el funcionamiento del control parental, es decir, en total se implementaron 63 endpoinds organizados en módulos. Pero

solo se presentan los más esenciales para el buen funcionamiento del sistema, en reposito de GitHub se encuentra más detallado, en específico la sección de backend: https://github.com/bryanarotingo-prog/Control-Parental-MT.git

app = FastAPI(title="Control Parental API", version="2.0")

El sistema se configuro como servicio del SO, es decir, que cuando se encienda o reinicie la PI 5, el sistema se iniciará automáticamente sin necesidad de encender manualmente por comandos.

[Service]

***ExecStart=/usr/bin/python3 -m uvicorn api_control:app ***

--host 0.0.0.0 --port 8000 --reload

Restart=always

EnvironmentFile=/home/mt69/.env

3.6.4.1 Inicialización de la API

El sistema inicia con la ejecución del backend desarrollado en el framework FastAPI, el cual se ejecuta automáticamente y expone los servicios definidos. Además, se importan las librerías necesarias para cumplir con lo dispuesto en el diseño.

Figura 25

Inicialización de la API



```
Users > MT69 > Pictures > api_control.py > ...
from fastapi import FastAPI, HTTPException
from pydantic import BaseModel
from dotenv import load_dotenv
import mysql.connector
import sqlite3
import os
import subprocess
import json
import datetime
import re
import threading
import time
import requests


import ipaddress

load_dotenv("/home/mt69/.env")

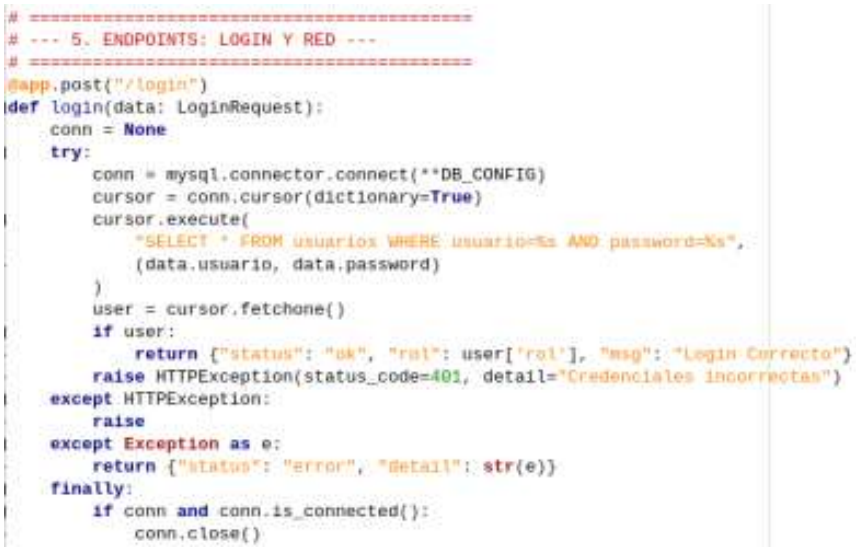
app = FastAPI(title="Control Parental API", version="2.0")

# ==========================================
# --- 1. CONFIGURACION DEL SISTEMA ---
# ==========================================
DB_CONFIG = {
    'host':     os.getenv('DB_HOST', 'localhost'),
    'user':     os.getenv('DB_USER', 'pi_admin'),
    'password': os.getenv('DB_PASSWORD', '1234'),
    'database': os.getenv('DB_NAME', 'control_parental'),
    'charset': 'utf8mb4'
}
```

La API se inicializa conectándose a la base de datos MariaDB y cargando las variables de entorno. Cuando el adminstrador ingresa las credenciales actúa este conforme al flujo de autenticación. Es decir, si el administrador ingresa mal las credenciales saldrá un mensaje de error y deberá volver a intentar. Una vez colocado las credenciales correctas, accede al dashboard de la aplicación.

Figura 26

Endpoint de autenticación



```
# ========================================
# --- 5. ENDPOINTS: LOGIN Y RED ---
# ========================================
@app.post("/login")
def login(data: LoginRequest):
    conn = None
    try:
        conn = mysql.connector.connect(**DB_CONFIG)
        cursor = conn.cursor(dictionary=True)
        cursor.execute(
            "SELECT * FROM usuarios WHERE usuario=%s AND password=%s",
            (data.usuario, data.password)
        )
        user = cursor.fetchone()
        if user:
            return {"status": "ok", "rol": user['rol'], "msg": "Login Correcto"}
        raise HTTPException(status_code=401, detail="Credenciales incorrectas")
    except HTTPException:
        raise
    except Exception as e:
        return {"status": "error", "detail": str(e)}
    finally:
        if conn and conn.is_connected():
            conn.close()
```

3.6.4.2 Endpoint de Configuración de Base de Datos del Sistema.

La comunicación entre la API y Pi-hole se llevó a cabo mediante dos mecanismos. El primero consiste en acceder directamente a la base de datos SQLite de Pi-hole, denominada gravity.db, donde se almacenan los dominios incluidos en la lista de bloqueo. Esto permite que la API consulte, agregue o elimine registros de manera directa según las acciones realizadas por el administrador. El segundo mecanismo utiliza comandos ejecutados dentro del contenedor Docker, lo que permite aplicar y actualizar los cambios realizados sin afectar el funcionamiento del resto del sistema.

Figura 27

Conexión de servicios y configuración del sistema

```
# ==========================================
# --- 1. CONFIGURACION DEL SISTEMA ---
# ==========================================
DB_CONFIG = {
    'host':     os.getenv('DB_HOST', 'localhost'),
    'user':     os.getenv('DB_USER', 'pi_admin'),
    'password': os.getenv('DB_PASSWORD', '1234'),
    'database': os.getenv('DB_NAME', 'control_parental'),
    'charset': 'utf8mb4'
}

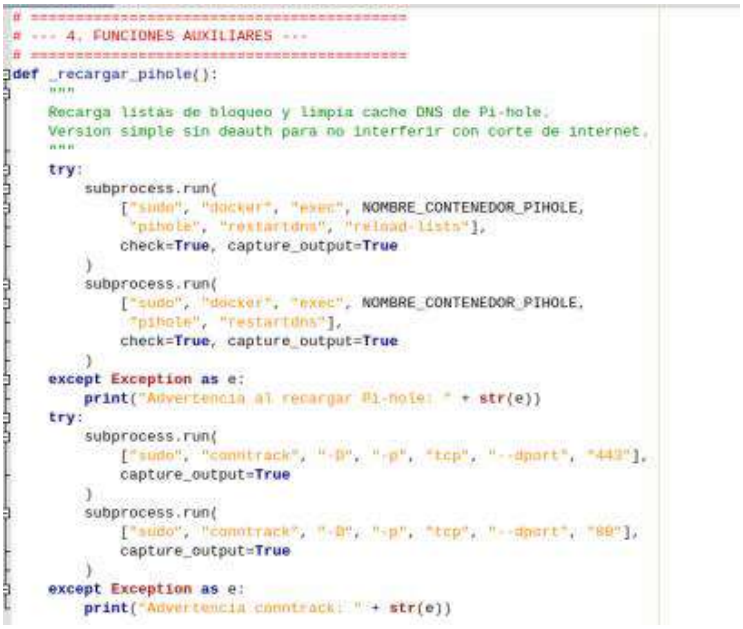
PIHOLE_DB             = os.getenv('PIHOLE_DB', '/home/mt69/pihole-v5/etc-pihole/gravity.db')
NOMBRE_CONTENEDOR_PIHOLE = os.getenv('PIHOLE_CONTAINER', 'pihole-v5')
PIHOLE_API_TOKEN      = os.getenv('PIHOLE_API_TOKEN', '')
GROQ_API_KEY          = os.getenv('GROQ_API_KEY', '')
SURICATA_LOG          = "/var/log/suricata/eve.json"
```

Cada vez que se agrega o elimina un dominio de la lista negra, la API recarga automáticamente Pi-hole para que los cambios se reflejen de inmediato. La API trabaja de forma integrada con dos bases de datos, la base de datos nativa de Pi-hole, utilizada para administrar las listas de bloqueo, y la base de datos del sistema desarrollada en MariaDB, donde se almacenan los usuarios, dispositivos, perfiles y demás información de la aplicación.

Se decidió mantener ambas bases de datos separadas, ya que unificar toda la información provocaba conflictos y no ofrecía una solución adecuada para la administración del sistema.

Figura 28

Recarga de listas de Pi-hole



```
# ========================================
# --- 4. FUNCIONES AUXILIARES ---
# ========================================
def _recargar_pihole():
    """
    Recarga listas de bloqueo y limpia cache DNS de Pi-hole.
    Version simple sin deauth para no interferir con corte de internet.
    """
    try:
        subprocess.run(
            ["sudo", "docker", "exec", NOMBRE_CONTENEDOR_PIHOLE,
             "pihole", "restartdns", "reload-lists"],
            check=True, capture_output=True
        )
        subprocess.run(
            ["sudo", "docker", "exec", NOMBRE_CONTENEDOR_PIHOLE,
             "pihole", "restartdns"],
            check=True, capture_output=True
        )
    except Exception as e:
        print("Advertencia al recargar Pi-hole: " + str(e))
    try:
        subprocess.run(
            ["sudo", "conntrack", "-D", "-p", "tcp", "--dport", "443"],
            capture_output=True
        )
        subprocess.run(
            ["sudo", "conntrack", "-D", "-p", "tcp", "--dport", "80"],
            capture_output=True
        )
    except Exception as e:
        print("Advertencia conntrack: " + str(e))
```

3.6.4.3 Endpoint de Bloqueo/Desbloqueo y Lista negra

El administrador puede hacer uso de la aplicación para que pueda mostrar las categorías y permitir activar o desactivar reglas de bloqueo sin que el usuario tenga que entrar directamente a Pi-hole. El sistema permite bloquear un sitio web ingresando únicamente una palabra clave, sin necesidad de escribir la dirección completa del dominio.

Además, el bloqueo o desbloqueo puede realizarse en cualquier momento desde la aplicación. Estos cambios se aplican a toda la red, por lo que todos los dispositivos conectados quedan sujetos a las mismas reglas de filtrado. Este proceso se lleva a cabo mediante el filtrado de consultas DNS realizado por Pi-hole.

Figura 29

Endpoint de lista negra



```python
@app.get("/listar-reglas")
def listar_reglas_pihole():
    """
    Lista la lista negra AGRUPADA por palabra clave.
    Cada grupo (comment) se muestra como una sola tarjeta con
    todos sus subdominios anidados.
    """
    try:
        conn   = sqlite3.connect(PIHOLE_DB)
        cursor = conn.cursor()
        cursor.execute("""
            SELECT comment, domain FROM domainlist
            WHERE type IN (1, 3) AND comment LIKE 'BLOQ_%'
            ORDER BY comment
        """)
        filas = cursor.fetchall()
        conn.close()

        grupos = {}
        for comment, domain in filas:
            palabra = comment.replace("BLOQ_", "", 1)
            if palabra not in grupos:
                grupos[palabra] = []
            grupos[palabra].append(domain)

        lista_negra = []
        for palabra, dominios in grupos.items():
            lista_negra.append({
                "palabra":    palabra,
                "categoria":  _detectar_categoria_grupo(palabra),
                "total":      len(dominios),
                "dominios":   sorted(dominios)
            })

        return {"status": "ok", "lista_negra": lista_negra, "total_grupos": len(lista_negra)}
    except Exception as e:
        return {"status": "error", "detail": str(e)}
```

3.6.4.4 Endpoint Bloqueo con Ayuda de IA

Cuando el administrador ingresa el nombre de una aplicación o sitio web, por ejemplo, TikTok, la API envía una consulta al modelo de lenguaje de Groq. A partir de esa consulta, el modelo identifica los dominios relacionados con el servicio, generando una lista que representa su huella digital en Internet.

Además, estos dominios se agregan automáticamente a la lista de bloqueo de Pi-hole, evitando que el administrador tenga que incorporarlos uno por uno. De esta manera, con solo ingresar una palabra es posible bloquear la mayoría de los dominios asociados a la aplicación o sitio web, logrando un filtrado más completo y eficiente.

Figura 30

Endpoint Bloqueo con IA



```python
@app.post("/redes/bloquear-con-ia/{app_nombre}")
def bloquear_app_con_ia(app_nombre: str):
    """Busca dominios con Groq IA y los bloquea en Pi-hole"""
    app_nombre = app_nombre.lower().strip()
    if app_nombre not in APPS_SOPORTADAS_IA:
        raise HTTPException(status_code=404,
            detail=f"App no soportada. Disponibles: {APPS_SOPORTADAS_IA}")

    comentario     = f"IA_{app_nombre.upper()}"
    dominios_ia    = _buscar_dominios_groq(app_nombre)
    dominios_base  = HUELLAS_DIGITALES.get(app_nombre, [])
    todos_dominios = list(set(dominios_ia + dominios_base))

    if not todos_dominios:
        return {"status": "error", "msg": "No se pudieron obtener dominios"}

    _quitar_dominios_pihole_comentario(comentario)
    agregados = _agregar_dominios_pihole(todos_dominios, comentario)
    _recargar_pihole()
    _enviar_notificacion(
        f"{app_nombre.upper()} bloqueado con IA",
        f"Se bloquearon {agregados} dominios de {app_nombre} usando IA.",
        "normal", "shield"
    )

    return {
        "status":           "ok",
        "app":              app_nombre,
        "bloqueado":        True,
        "dominios_ia":      len(dominios_ia),
        "dominios_base":    len(dominios_base),
        "total_bloqueados": agregados,
        "dominios":         todos_dominios,
        "msg":              f"{app_nombre.upper()} bloqueado con {agregados} dominios via IA"
    }
```

3.6.4.5 Endpoint de Control de Horario de los Dispositivos

El control de internet por dispositivo combina la base de datos con las reglas iptables. Este código corresponde al endpoint que permite bloquear o restaurar el acceso a Internet de un dispositivo mediante reglas de iptables. Cuando el padre hace uso de la sección de horarios o quitar el internet a cada uno de los dispositivos se aplican estas reglas ya definidas en el API.

Figura 31

Endpoint de control de dispositivos



```python
@app.post("/dispositivos/internet")
def control_internet_mac(data: InternetRequest):
    """
    Corta o restaura internet por MAC address.
    Mas confiable que por IP porque la IP puede cambiar.
    """
    try:
        mac = data.mac.upper() if hasattr(data, 'mac') else None
        conn = mysql.connector.connect(**DB_CONFIG)
        cursor = conn.cursor(dictionary=True)

        if mac:
            cursor.execute(
                "SELECT ip, apodo FROM dispositivos WHERE mac = %s", (mac,)
            )
        else:
            cursor.execute(
                "SELECT ip, apodo FROM dispositivos WHERE ip = %s",
                (data.ip,)
            )

        dispositivo = cursor.fetchone()
        if not dispositivo:
            conn.close()
            return {"status": "error", "msg": "Dispositivo no encontrado"}

        ip = dispositivo["ip"]
        apodo = dispositivo["apodo"] or ip

        if data.accion == "off":
            subprocess.run(
                ["sudo", "iptables", "-D", "FORWARD", "-s", ip, "-j", "DROP"],
                capture_output=True
            )
            subprocess.run(
                ["sudo", "iptables", "-I", "FORWARD", "3", "-s", ip, "-j", "DROP"],
                check=True, capture_output=True
            )
            cursor.execute(
                "UPDATE dispositivos SET internet = 0 WHERE ip = %s", (ip,)
            )
            conn.commit()
            conn.close()
            return {
                "status": "ok",
                "msg": f"Internet CORTADO a {apodo} ({ip})"
            }

        elif data.accion == "on":
            subprocess.run(
```

3.6.4.6 Endpoint de Generación de Reportes con IA

El sistema permite generar reportes sobre la actividad de la red a partir de la información recopilada de los dispositivos conectados a la red y de los servicios que utilizan. Con estos datos,

el backend prepara un prompt específico según el tipo de reporte y lo envía al modelo de lenguaje Llama 3.3 70B Versatile mediante la API de Groq. El modelo analiza la información recibida y genera un resumen del comportamiento de la red, identificando aspectos relevantes de la navegación. Para que, por consiguiente, la respuesta se devuelve en formato JSON para que la aplicación la procese y presente al usuario de forma clara y organizada. Esto se guarda en la base de datos siempre y cuando se genere el reporte.

Figura 32

Endpoint de generación de reportes con IA

```
@app.get("/reporte/generar")
def generar_reporte_completo():
    """Genera reporte completo con analisis heuristico + IA + PDF"""
    try:
        hoy      = datetime.datetime.now().strftime("%Y-%m-%d")
        hora_act = datetime.datetime.now().strftime("%H:%M:%S")

        stats      = _obtener_stats_pihole()
        heuristico = _analisis_heuristico_logs()
        top        = _obtener_top_bloqueados()

        top_str  = str(list(top["top_bloqueados"].keys())[:10]) if top["top_bloqueados"] else "Ninguno"
        apps_str = ", ".join(heuristico.get("apps_detectadas", [])) or "Ninguna"

        prompt = f"""
Eres Pachamama, una IA protectora para padres de familia.
Habla de forma calida, clara y sin tecnicismos.
Se creativo y no repitas siempre lo mismo.

DATOS DEL DIA:
- Consultas DNS: {stats['total_consultas']}
- Dominios bloqueados: {stats['bloqueados']} ({stats['porcentaje_bloqueo']:.1f}%)
- Apps detectadas: {apps_str}
- Hora de mayor actividad: {heuristico.get('hora_pico', 'N/A')}
- Top bloqueados: {top_str}
- Comportamiento sospechoso: {'Si' if heuristico.get('comportamiento_sospechoso') else 'No'}
- Dispositivos conectados: {stats['clientes_activos']}

Devuelve UNICAMENTE un JSON valido sin tildes ni caracteres especiales:
{{
    "nivel_riesgo": "Bajo, Medio, Alto o Critico",
    "resumen": "Resumen breve del dia en 2 oraciones",
    "sitios_frecuentes": "Que sitios normales visitaron",
    "patron_horario": "A que horas hubo mas actividad",
    "sitios_bloqueados": "Que se bloqueo hoy",
    "amenazas_red": "Hubo algo preocupante o no",
    "recomendaciones": "2 o 3 consejos practicos para los padres"
}}
"""
        headers = {"Authorization": f"Bearer {GROQ_API_KEY}", "Content-Type": "application/json"}
        payload = {
            "model":    "llama-3.3-70b-versatile",
            "messages": [{"role": "user", "content": prompt}],
            "response_format": {"type": "json_object"}
        }
        resp_ia = requests.post(
            "https://api.groq.com/openai/v1/chat/completions",
            json=payload, headers=headers, timeout=30
        )
```

Además de ello, se implementó un servicio para consultar el historial de reportes almacenados en MariaDB. Este recupera los diez reportes más recientes, lo que incluirá información detallada de las diferentes actividades realizadas dentro de la red. Con esto se puede

realizar un el reporte general, que se lo imprime en formato PDF para una mejor apreciación para el administrador.

Figura 33

Generar reporte como PDF

```python
@app.get("/reporte/pdf/{nombre}")
def descargar_pdf(nombre: str):
    """Descarga el PDF generado"""
    from fastapi.responses import FileResponse
    ruta = f"/tmp/{nombre}"
    if not os.path.exists(ruta):
        raise HTTPException(status_code=404, detail="PDF no encontrado")
    return FileResponse(ruta, media_type="application/pdf", filename=nombre)


@app.get("/reporte/historial")
def historial_reportes():
    """Ultimos 10 reportes guardados en MariaDB"""
    try:
        conn   = mysql.connector.connect(**DB_CONFIG)
        cursor = conn.cursor(dictionary=True)
        cursor.execute("""
            SELECT id, fecha, hora, total_consultas, anuncios_bloqueados,
                   porcentaje_bloqueo, nivel_riesgo, resumen, recomendaciones
            FROM reportes_ia
            ORDER BY fecha DESC, hora DESC
            LIMIT 10
        """)
        reportes = cursor.fetchall()
        conn.close()
        for r in reportes:
            r["fecha"] = str(r["fecha"])
            r["hora"]  = str(r["hora"])
            r["porcentaje_bloqueo"] = float(r["porcentaje_bloqueo"])
        return {"status": "ok", "reportes": reportes}
    except Exception as e:
        return {"status": "error", "detail": str(e)}
```

3.6.5 Módulo de Monitoreo e Interfaz de Usuario

Para ofrecer una solución más completa, se desarrollaron dos aplicaciones móviles: una destinada al administrador y otra para el dispositivo que será monitoreado.

La aplicación del administrador fue creada mediante React Native como framework principal y Expo SDK 56 como plataforma de desarrollo. Esta unión de herramientas hizo posible el desarrollo de una aplicación nativa multiplataforma para dispositivos Android e iOS a partir de un único código fuente realizado en JavaScript.

En cambio, la aplicación del dispositivo monitoreado estuvo desarrollada mediante Android Studio, dado que React Native tenía limitaciones en el acceso e implementación de algunas características de seguridad propias del sistema operativo Android. Debido a estas restricciones, se optó por el desarrollo nativo, lo que permitió integrar de forma adecuada las funcionalidades requeridas para el monitoreo y control del dispositivo.

Figura 34

Dependencias de la app del administrador

Tecnología	Versión	Función
React Native	0.85.3	Framework base de la app
Expo SDK	56	Herramientas de desarrollo y compilación
React Navigation	7.1	Navegación entre pantallas
Axios	1.17.0	Comunicación HTTP con el backend
AsyncStorage	56	Notificaciones push
Expo Device	56	Información del dispositivo

3.6.5.1 Navegación y Estructura Principal

La aplicación utiliza una combinación de navegación por pestañas inferiores denominada Bottom Tab Navigator, para las pantallas principales y navegación en pila Stack Navigator para las pantallas secundarias.

Figura 35

Navegación y estructura principal de la aplicación

```
import React, { useEffect, useRef, useState, createContext, useContext } from 'react';
import { NavigationContainer }          from '@react-navigation/native';
import { createBottomTabNavigator }     from '@react-navigation/bottom-tabs';
import { createNativeStackNavigator }   from '@react-navigation/native-stack';
import {
  Text, View, Platform, TouchableOpacity,
  Animated, Alert
} from 'react-native';
import AsyncStorage                     from '@react-native-async-storage/async-storage';
import * as Notifications               from 'expo-notifications';
import * as Device                      from 'expo-device';

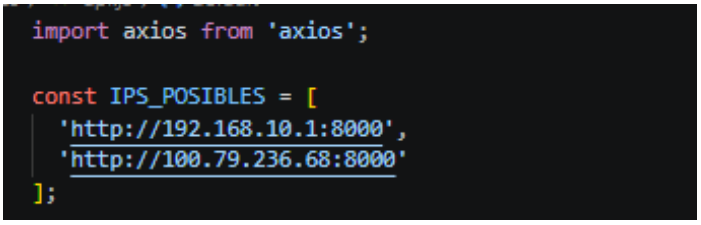
import LoginScreen        from './screens/LoginScreen';
import DispositivosScreen from './screens/DispositivosScreen';
import RedesScreen        from './screens/RedesScreen';
import HorariosScreen     from './screens/HorariosScreen';
import AlertasScreen      from './screens/AlertasScreen';
import ReportesScreen     from './screens/ReportesScreen';
import ConfigScreen       from './screens/ConfigScreen';
import ChildGuardScreen   from './screens/ChildGuardScreen';
import { getAlertasNuevas } from './services/api';
```

3.6.5.2 Comunicación con el Backend

Toda la comunicación entre la aplicación móvil y el servidor se maneja desde un solo archivo llamado services/api.js. La aplicación primero intenta conectarse directamente a la red local WiFi, donde se encuentra el servidor. Si no logra acceder porque el usuario está fuera de casa, utiliza la conexión segura mediante VPN Tailscale para comunicarse de forma remota.

Figura 36

Direcciones para la conexión local o externa



Una vez establecida la conexión con el servidor, el usuario podrá iniciar sesión para acceder al panel principal de la aplicación. Antes de permitir el acceso, el sistema verifica que exista comunicación con el servidor. Si el dispositivo no se encuentra conectado a la red local y tampoco puede acceder mediante Tailscale, se muestra un mensaje indicando que debe establecer la conexión correspondiente antes de iniciar sesión.

También si el usuario ofrece credenciales incorrectas, la aplicación informa que el nombre de usuario o la contraseña son incorrectos. Cuando la conexión se establece correctamente, la aplicación presenta el estado de conexión con un indicador de color y la dirección IP utilizada para conectarse a través del servidor, ya sea la dirección de red local o dirección asignada por Tailscale. Esto proporciona al usuario una rápida comprensión del tipo de conexión utilizado.

Figura 37

Login de la APP



3.6.5.3 Dispositivos Conectados a la Red

La sección Dispositivos Conectados muestra los equipos conectados al punto de acceso de la Raspberry Pi 5. La API obtiene y actualiza automáticamente la información de cada dispositivo, como IP, nombre y fabricante, y la combina con el nivel de riesgo calculado. Desde la aplicación, los padres pueden identificar cada equipo, asignar un nombre, bloquear o restablecer el acceso a Internet y supervisar su actividad en tiempo real.

Figura 38

Screen de dispositivos conectados

```
import React, { useState, useEffect, useCallback } from 'react';
import {
  View, Text, TouchableOpacity, StyleSheet, Alert,
  TextInput, Modal, ActivityIndicator, RefreshControl, ScrollView
} from 'react-native';
import {
  getDispositivosActivos, cortarInternet, asignarApodo,
  getDispositivosBloqueados, darAccesoDispositivo,
  descartarDispositivo, readmitirDispositivo, getScoresHeuristicos,
  getHorarioNocturno, setHorarioNocturno
} from '../services/api';

const C = {
  bg:'#E8F4FD', surface:'#FFFFFF', surfaceAlt:'#F0F8FF',
  primary:'#1a92d2', danger:'#E53935', warning:'#F57C00',
  success:'#2E7D32', purple:'#5E35B1', textPrimary:'#1a1a2e',
  textSecondary:'#37474F', textMuted:'#78909C', border:'#B3D9F0',
};

const NIVEL_COLOR = { Bajo: '#2E7D32', Medio: '#F57C00', Alto: '#E53935' };
const NIVEL_BG    = { Bajo: '#F1F8E9', Medio: '#FFF8E1', Alto: '#FFEBEE' };
const NIVEL_BORDER= { Bajo: '#C5E1A5', Medio: '#FFE082', Alto: '#FFCDD2' };
const NIVEL_EMOJI = { Bajo: '✅', Medio: '⚠️', Alto: '🔴' };

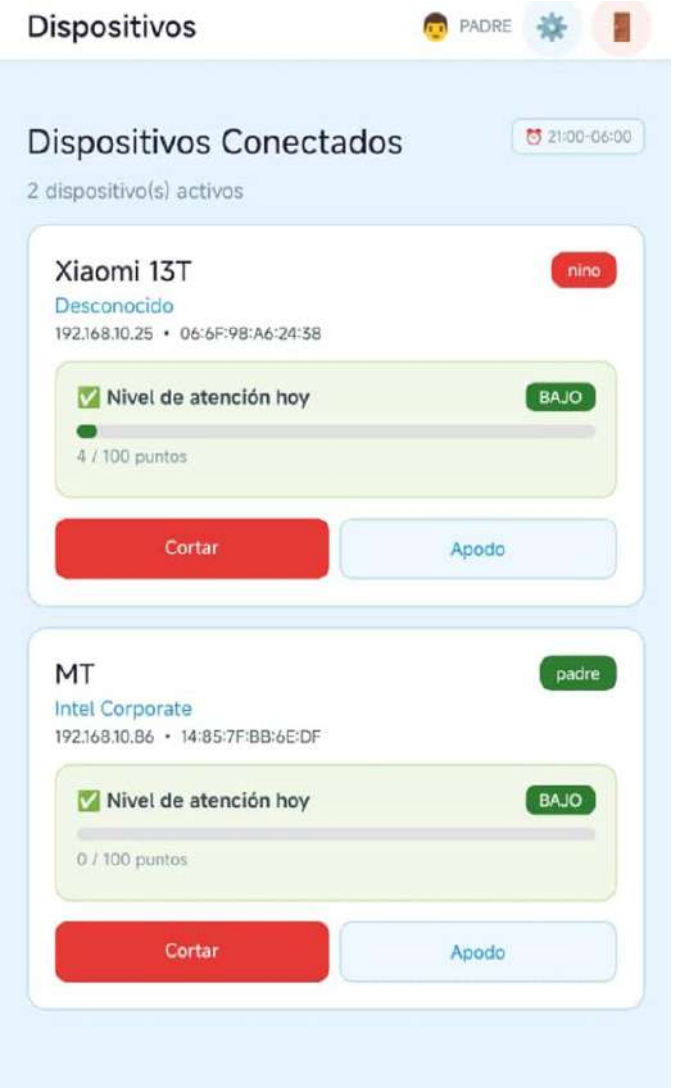
function ScoreCard({ score }) {
  if (!score) return null;
  const color  = NIVEL_COLOR[score.nivel] || C.textMuted;
  const bg     = NIVEL_BG[score.nivel]    || '#F5F5F5';
  const border = NIVEL_BORDER[score.nivel]|| '#E0E0E0';
  const emoji  = NIVEL_EMOJI[score.nivel] || '❓';
  const pct    = Math.min(score.score, 100);

  const barColor = score.nivel === 'Alto' ? C.danger
                 : score.nivel === 'Medio' ? C.warning
                 : C.success;
```

En la aplicación, la información se presenta de forma clara y organizada para facilitar su comprensión. Cada dispositivo se muestra en una tarjeta con sus datos principales, el nivel de riesgo y las acciones disponibles. De esta manera, los padres pueden identificar rápidamente el estado de cada equipo, analizar su actividad con solo observar la pantalla y tomar decisiones sin necesidad de conocimientos técnicos.

Figura 39

Screen de dispositivos conectados



3.6.5.4 Sistema de Notificaciones en Tiempo Real

La aplicación implementa un sistema de polling cada 15 segundos que consulta al servidor si hay alertas nuevas y las muestra como notificaciones push nativas. En esta parte hay que revisar

a fondo ya que detrás de las notificaciones hay una estructura que clasifica el tráfico y enviara alertas si es necesario.

Figura 40

Sistema de notificaciones en tiempo real

```
const iniciarPolling = () => {
  pollingRef.current = setInterval(async () => {
    try {
      const notifActivas = await AsyncStorage.getItem('notificaciones_activas');
      if (notifActivas === 'false') return;
      const resp = await getAlertasNuevas();
      if (!resp.data.hay_nuevas || !resp.data.alertas.length) return;
      for (const alerta of resp.data.alertas) {
        if (idsNotificados.current.has(alerta.id)) continue;
        idsNotificados.current.add(alerta.id);
        await Notifications.scheduleNotificationAsync({
          content: {
            title: `Control Parental — ${alerta.dispositivo}`,
            body:  alerta.mensaje,
            data:  { alerta },
          },
          trigger: null,
        });
      }
      if (idsNotificados.current.size > 500) idsNotificados.current.clear();
    } catch {}
  }, 15000);
};
```

En la aplicación, las notificaciones y alertas se presentan de forma clara para informar a los padres sobre los eventos que ocurren en la red y en los dispositivos supervisados. Estas alertas pueden indicar, por ejemplo, intentos de acceso a páginas bloqueadas, actividad fuera del horario permitido o la detección de sitios considerados de riesgo.

Además, el sistema incorpora parámetros previamente definidos para reducir las falsas alertas ocasionadas por conexiones automáticas de aplicaciones que se ejecutan en segundo plano, mostrando únicamente las notificaciones que realmente requieren la atención del administrador

Figura 41

Sección de alertas de la app



3.6.5.5 Aplicación del Cliente Monitoreado

La aplicación ChildGuard fue desarrollada de forma nativa para Android utilizando el lenguaje de programación Kotlin y el entorno de desarrollo Android Studio. Esta aplicación sirve para realizar el seguimiento de cada dispositivo internamente, es decir, se puede revisar cuantas aplicaciones tiene el dispositivo, cuales usan más tiempo, poder bloquear esas aplicaciones, entre otras, esta se encuentra como una beta debido a su limitación para Android. En la siguiente tabla se presenta las tecnologías usadas en esta aplicación.

Tabla 17

Tecnologías de la aplicación de cliente ChildGuard

Tecnología	Versión	Función
OkHttp3	4.12.0	Comunicación HTTP con la Pi
Gson	2.10.1	Serialización/deserialización JSON
Kotlinx Coroutines	1.7.3	Tareas en segundo plano asíncronas
WorkManager	2.9.0	Tareas programadas en background
AppCompat	1.6.1	Compatibilidad con versiones antiguas
Material Design	1.11.0	Componentes visuales de Google

Se emplearon las APIs de accesibilidad de Android en si AccessibilityService para detectar y bloquear aplicaciones en tiempo real, y UsageStatsManager para obtener estadísticas de uso. La comunicación con el servidor se implementó mediante la librería OkHttp3, realizando peticiones HTTP al backend FastAPI de la Raspberry Pi cada 60 segundos para sincronizar el estado de las aplicaciones bloqueadas y reportar la actividad del dispositivo.

De esa manera se desarrolló la aplicación, algo minimalista y simple ya que la única cuestión era que debe conectarse al sistema para que interactúe con la aplicación del padre. Toda la codificación y configuraciones se encuentran en el repositorio más detallado: https://github.com/bryanarotingo-prog/Control-Parental-MT/tree/childguard

Además de la codificación de la aplicación, fue necesario implementar permisos especiales para garantizar su correcto funcionamiento. Estos permisos deben ser habilitados manualmente por el usuario antes de utilizar la aplicación, ya que permiten acceder a funciones del sistema operativo indispensables para el monitoreo y control del dispositivo. De esta manera, la aplicación puede interactuar de forma adecuada con Android y ejecutar sus procesos de manera continua y sincronizada.

<uses-permission android:name="android.permission.PACKAGE_USAGE_STATS"/>

<uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>

<uses-permission android:name="android.permission.FOREGROUND_SERVICE_DATA_SYNC"/>

<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>

<uses-permission android:name="android.permission.BIND_ACCESSIBILITY_SERVICE"/>

Todos estos permisos hacen posible que la aplicación supervise el uso del dispositivo, permanezca activa, se reinicie automáticamente tras un reinicio del teléfono y aplique las restricciones configuradas por el administrador. Toda esta información la encontramos en el repositorio en específico en la sección: app/src/main/AndroidManifest.xml.

3.6.5.5.1 Vinculación con la Aplicación del Administrador

Cuando se haya otorgado el acceso y los permisos correspondientes a la aplicación Childguard, se debe realizar la vinculación esto se logró utilizando un código de 6 caracteres que la aplicación del administrador lo genera. Cuando se ingresa el código se registra automáticamente en el backend con un identificador único. La implementación completa se encuentra en el repositorio en la sección de la aplicación childguard: MainActivity.kt.

Figura 42

Sección de la vinculación con la aplicación del administrador

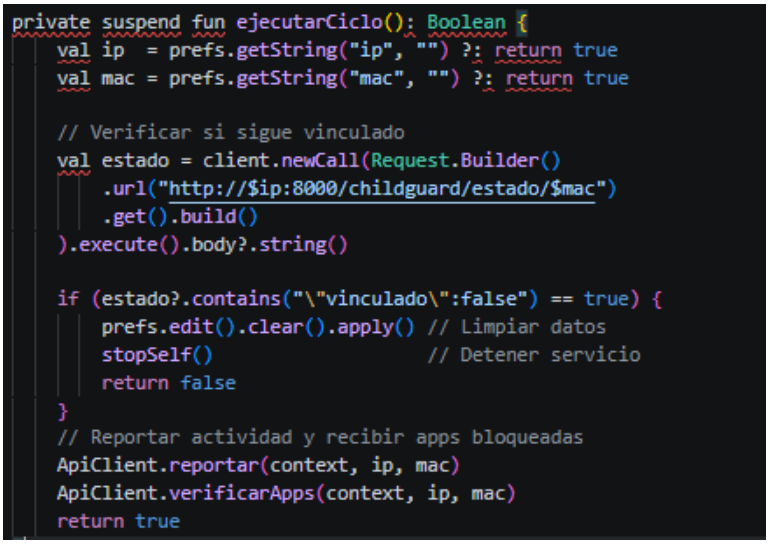
```
private fun vincularDispositivo(ip: String, codigo: String, nombre: String) {
    val mac = Settings.Secure.getString(contentResolver,
                                        Settings.Secure.ANDROID_ID)
    // Registrar en la Pi
    val bodyReg = gson.toJson(mapOf(
        "mac"         to mac,
        "dispositivo" to nombre,
        "token"       to codigo
    ))
    client.newCall(Request.Builder()
        .url("http://$ip:8000/childguard/registrar")
        .post(RequestBody.create(mediaType, bodyReg))
        .build()
    ).execute()
```

3.6.5.5.2 Servicio en Segundo Plano

Para evitar que la aplicación deje de funcionar y garantice sus funciones principales, se debe ejecutar un foreground service, esto garantiza que los dispositivos Android no elimine la memoria. Para ello se realiza una sincronización con el servidor cada 60 segundos. Todo esto esta desglosado dentro del archivo GuardService.kt.

Figura 43

Sección del servicio en segundo plano de Childguard



```
private suspend fun ejecutarCiclo(): Boolean {
    val ip  = prefs.getString("ip", "") ?: return true
    val mac = prefs.getString("mac", "") ?: return true

    // Verificar si sigue vinculado
    val estado = client.newCall(Request.Builder()
        .url("http://$ip:8000/childguard/estado/$mac")
        .get().build()
    ).execute().body?.string()

    if (estado?.contains("\"vinculado\":false") == true) {
        prefs.edit().clear().apply() // Limpiar datos
        stopSelf()                   // Detener servicio
        return false
    }
    // Reportar actividad y recibir apps bloqueadas
    ApiClient.reportar(context, ip, mac)
    ApiClient.verificarApps(context, ip, mac)
    return true
```

3.6.5.5.3 Bloqueo de Aplicaciones

Para la parte de bloqueo de aplicaciones se implementó un permiso de Android en este acaso AccessibilityService, aplicaciones como Anydesk hacen uso de este acceso. Este acceso nos sirve para monitorear de manera constante que aplicación está en primer plano. Desde la aplicación del padre hay la opción de bloquear aplicaciones dentro del dispositivo Android. Cuando el hijo intente abrir una app bloqueada, el sistema la cierra inmediatamente. Esto se logró con el acceso y los permisos de Android que se codifica en el archivo AppBlockerService.kt.

Figura 44

Sección de bloqueo de aplicaciones mediante AccessibilityService



```
override fun onAccessibilityEvent(event: AccessibilityEvent?) {
    if (event?.eventType != AccessibilityEvent.TYPE_WINDOW_STATE_CHANGED) return
    val packageName = event.packageName?.toString() ?: return
    val appsB = prefs.getStringSet("apps_bloqueadas", emptySet())

    if (appsB.contains(packageName)) {
        // Registrar intento bloqueado para reportar al padre
        val intentos = prefs.getStringSet("intentos_bloqueadas",
                       mutableSetOf())?.toMutableSet() ?: mutableSetOf()
        intentos.add(packageName)
        prefs.edit().putStringSet("intentos_bloqueadas", intentos).apply()

        // Cerrar la app y volver al inicio
        performGlobalAction(GLOBAL_ACTION_HOME)

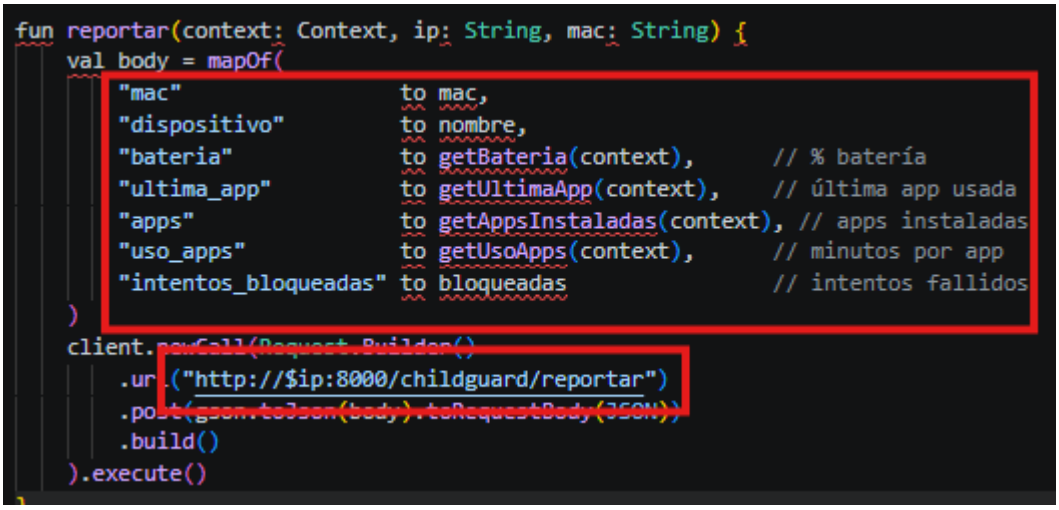
        // Notificar al hijo
        Toast.makeText(this,
            "\"$nombre\" está bloqueada por tus padres",
            Toast.LENGTH_LONG).show()
    }
```

3.6.5.5.4 Reporte de actividad al servidor

En el archivo ApiClient se encarga de realizar la gestión de recopilar la información de cada uno de los dispositivos y las envía al servidor cada 60 segundos, esto se lo hace para que no se sature con varios servicios dentro del backend, además de tener un tiempo prudente para actualizar la información. Todo este bloque implementado se lo puede encontrar en el archivo ApiCliente.kt.

Figura 45

Sección del reporte de actividad al servidor



3.6.5.5.5 Inicio automático al Encender el Celular

Para garantizar que todas las funciones descritas anteriormente se mantengan operativas, fue necesario implementar el servicio BroadcastReceiver. Este componente permite que la aplicación reciba eventos del sistema, como el apagado o el reinicio del dispositivo, y ejecute automáticamente las acciones necesarias para restablecer los servicios de monitoreo.

De esta manera, las configuraciones y restricciones enviadas desde la aplicación del administrador permanecen activas, asegurando el funcionamiento continuo del sistema sin que el usuario tenga que intervenir.

Figura 46

Sección del inicio automático al encender el celular

```
class BootReceiver : BroadcastReceiver() {
    override fun onReceive(context: Context, intent: Intent) {
        if (intent.action == Intent.ACTION_BOOT_COMPLETED) {
            GuardService.start(context)
        }
    }
}
```

Siguiendo con lo mencionado anteriormente, todos estos archivos principales ayudan a que la aplicación tenga una sincronización con el sistema operativo y con el sistema del control parental. Todo esto siguiendo los pasos correctamente, para que así la aplicación se vincule sin ningún problema y pueda ser monitoreado todo el tiempo.

Figura 47

Interfaz de la aplicación de Childguard



3.6.6 Módulo de Análisis Heurístico

El sistema de control parental cuenta con un mecanismo de análisis que revisa de forma continua la actividad de navegación de los dispositivos conectados a la red. A diferencia de un sistema que solo bloquea páginas mediante listas predefinidas, este también considera aspectos como la frecuencia con la que se accede a ciertos sitios, los horarios de uso y otros patrones de navegación. Con esta información, asigna a cada dispositivo un puntaje de riesgo, denominado Score, que va de 0 a 100 y permite identificar el nivel de riesgo asociado a su comportamiento en Internet.

3.6.6.1 Análisis de Movimientos en la Red

Esta primera función se desglosa del diseño propuesto anteriormente, la cual realiza la acción de revisar las ultimas 500 líneas del log de Pi-hole para determinar patrones de navegación. Toda esta información se consolida para que luego salga con más información de los demás servicios. Toda la codificación se encuentra en el archivo api_control.py del sistema que se encuentra en el repositorio en la sección de módulo de análisis heurístico:

Figura 48

Sección de detección de patrones

```
1499  def _analisis_heuristico_logs() -> dict:
1500      """Analiza el log de Pi-hole para detectar patrones"""
1501      resultado = {
1502          "intentos_bloqueados": [],
1503          "apps_detectadas": set(),
1504          "horas_actividad": {},
1505          "comportamiento_sospechoso": [],
1506          "total_lineas": 0
1507      }
1508
1509  >   apps_keywords = {···
1520
1521  >   try:···
1556
```

Figura 49

Sección de lectura del registro de Pi hole

```
1521    try:
1522        proc = subprocess.Popen(
1523            ["sudo", "docker", "exec", NOMBRE_CONTENEDOR_PIHOLE,
1524             "tail", "-n", "500", "/var/log/pihole/pihole.log"],
1525            stdout=subprocess.PIPE, stderr=subprocess.PIPE
1526        )
1527        lineas = proc.stdout.read().decode("utf-8", errors="ignore").split("\n")
1528        resultado["total_lineas"] = len(lineas)
1529
1530        for linea in lineas:
1531            if not linea.strip():
1532                continue
```

Como se mencionó anteriormente el algoritmo obtiene las últimas entradas del archivo de registro generado por Pi-hole. A partir de esta información se analizan las consultas DNS realizadas por los dispositivos conectados y determina las acciones.

3.6.6.2 Identificación de Aplicaciones Prohibidas

Además de ello, hay una sección para identificar aplicaciones con VPN o herramientas similares para evadir el control parental. Si detecta alguno de estos intentos, lo marca como sospechoso y lo guarda para que pueda ser revisado posteriormente.

Figura 50

Sección de identificación de aplicaciones con VPN

```
            if "blacklisted" in linea:
                if any(w in linea.lower() for w in ["vpn", "proxy", "tunnel", "tor"]):
                    resultado["comportamiento_sospechoso"].append(linea.strip()[:100])
```

3.6.6.3 Control de Hora de Navegación

Finalmente, el sistema realiza un seguimiento de cada dispositivo para conocer el tiempo que permanece navegando en la red. Toda esta información se recopila y analiza para determinar si el comportamiento del dispositivo se encuentra dentro de los parámetros permitidos. Con base en estos resultados, el administrador puede tomar decisiones y realizar los cambios necesarios en la configuración del control parental.

Figura 51

Horarios de navegación

```
if resultado["horas_actividad"]:
    hora_pico = max(resultado["horas_actividad"], key=resultado["horas_actividad"].get)
    resultado["hora_pico"] = f"{hora_pico}:00"
else:
    resultado["hora_pico"] = "Sin datos"

return resultado
```

3.6.6.4 Puntaje de Cada Dispositivo

El sistema genera automáticamente una sección dentro del reporte PDF y una pequeña parte dentro de la sección de dispositivos conectados, sobre el análisis heurístico de cada dispositivo monitoreado. En esta sección se muestra el nivel de riesgo obtenido, el puntaje asignado y diferentes indicadores relacionados con el comportamiento de navegación.

Figura 52

Sección del análisis para el puntaje de los dispositivos



```
# Seccion: Analisis Heuristico por dispositivo
try:
    import mysql.connector as _mc2
    conn_h = _mc2.connect(**DB_CONFIG)
    cur_h  = conn_h.cursor(dictionary=True)
    cur_h.execute("""
        SELECT dispositivo, score, nivel,
               alertas_red_social, alertas_adulto,
               alertas_apuestas, alertas_peligroso, alertas_nocturnas
        FROM scores_heuristicos
        WHERE fecha = CURDATE()
        ORDER BY score DESC
    """)
    scores_pdf = cur_h.fetchall()
    conn_h.close()
    if scores_pdf:
        elementos.append(Paragraph("Analisis Heuristico por Dispositivo", seccion_style))
        elementos.append(Paragraph(
            "Evaluacion del comportamiento de navegacion de cada dispositivo "
            "en las ultimas 24 horas. Considera intentos a contenido prohibido, "
            "sitios peligrosos, apuestas, redes sociales y actividad nocturna (21:00-06:00).",
            normal_style
        ))
        elementos.append(Spacer(1, 0.3*cm))
        NIVEL_COLOR_PDF = {"Bajo":"#27ae60","Medio":"#f39c12","Alto":"#e74c3c"}
        for s in scores_pdf:
            disp  = s["dispositivo"]
            score = s["score"]
            nivel = s["nivel"]
            color_nivel = colors.HexColor(NIVEL_COLOR_PDF.get(nivel, "#555555"))
            elementos.append(Paragraph(
                "<b>Dispositivo: " + disp + "</b>",
                ParagraphStyle("dt", parent=estilos["Normal"],
                    fontSize=11, textColor=colors.HexColor("#1a5276"),
                    spaceBefore=10, spaceAfter=4)
            ))
            filas_disp = [
                ["Metrica", "Valor", "Evaluacion"],
```

La pantalla de dispositivos muestra en tiempo real el nivel de riesgo de cada dispositivo calculado por el servidor, con indicadores visuales codificados por color, para ello se realizó un umbral para determinar el comportamiento de cada dispositivo.

El sistema utiliza umbrales para evitar generar alertas por conexiones automáticas que realizan las aplicaciones o las páginas web en segundo plano. De esta manera, no todos los intentos de acceso se consideran una acción del usuario. Por ejemplo, en las categorías de contenido para adultos, sitios de apuestas y redes sociales bloqueadas, el sistema espera varios intentos antes de aumentar el puntaje de riesgo. En cambio, los accesos a sitios considerados peligrosos se registran desde el primer intento, ya que representan un posible riesgo para la seguridad de la red.

Figura 53

Sección de reportes y análisis heurístico



Clasificación de riesgos:

- Cuando el puntaje se encuentra entre 0 y 15, el dispositivo se clasifica con un riesgo bajo y se muestra con un indicador de color verde. Esto significa que la actividad registrada es normal y no es necesario aplicar restricciones adicionales.
- Si el puntaje está entre 16 y 40, el dispositivo se clasifica con un riesgo medio y se identifica con un indicador de color naranja. En este caso, se recomienda que los padres revisen las alertas generadas y den seguimiento a la actividad del dispositivo.

- Cuando el puntaje es de 41 a 100, el dispositivo se clasifica con un riesgo alto y se muestra con un indicador de color rojo. Esto indica que se detectó una mayor cantidad de actividades de riesgo, por lo que se recomienda aplicar restricciones de acceso a Internet o supervisar más de cerca el uso del dispositivo.

3.6.6.5 Análisis del tráfico de red

En este caso suricata actúa como un inspector de todo el tráfico de la red del Hostpod a la que los diferentes dispositivos están conectados. El sistema usa las instancias de los logs de Suricata en este caso las ultimas 50 líneas.

proc = subprocess.Popen(['tail', '-n', '50', SURICATA_LOG], stdout=subprocess.PIPE)

El sistema agrupa datos depurados para clasificar con precisión el riesgo mediante Inteligencia Artificial, almacenando de forma optimizada los resultados en la base de datos. Toda esta información también se hará visible en la parte de reportes.

Figura 54

Reporte del Análisis Heurístico



Sitios frecuentes: youtube, instagram, twitter, facebook, whatsapp

Patron de uso: La hora de mayor actividad fue a las 15:00

Bloqueos del dia: mobile.pipe.aria.microsoft.com, firebaselogging.googleapis.com, api.ad.intl.xiaomi.com, app-measurement.com, yt3.ggpht.com, crashlyticsreports-pa.googleapis.com, i.ytimg.com

Recomendaciones: Verifica los permisos de las aplicaciones, establece limites de tiempo en las redes sociales y habla con tus hijos sobre la importancia de la seguridad en linea

Analisis Heuristico por Dispositivo

Evaluacion del comportamiento de navegacion de cada dispositivo en las ultimas 24 horas. Considera intentos a contenido prohibido, sitios peligrosos, apuestas, redes sociales y actividad nocturna (21:00-06:00).

Dispositivo: Xiaomi 13T

Metrica	Valor	Evaluacion
Nivel de riesgo	Bajo	Score: 0/100
Intentos redes sociales	1	Normal
Intentos contenido adulto	0	Bajo
Intentos apuestas	0	Normal
Intentos sitios peligrosos	0	Sin incidentes
Actividad nocturna (21-06h)	0	Normal

Dispositivo: Xiaomi-13T

Metrica	Valor	Evaluacion
Nivel de riesgo	Bajo	Score: 0/100
Intentos redes sociales	0	Normal
Intentos contenido adulto	0	Bajo
Intentos apuestas	0	Normal
Intentos sitios peligrosos	0	Sin incidentes
Actividad nocturna (21-06h)	0	Normal

Estado de la Red y Amenazas Detectadas (Suricata IDS)

El sistema de deteccion de intrusiones Suricata monitorea el trafico de red en tiempo real con 51894 reglas activas. A continuacion se presentan los eventos de seguridad detectados durante el periodo analizado, complementando el analisis heuristico de comportamiento.

Estado general de la red: **TRAFICO NORMAL**

Tipo de evento: Trafico normal

Que esta pasando en tu red: Evento de Suricata IDS que indica trafico de aplicacion HTTP en direccion incorrecta, sin embargo se identifico como Go HTTP Client, lo que sugiere trafico normal

Acciones recomendadas:

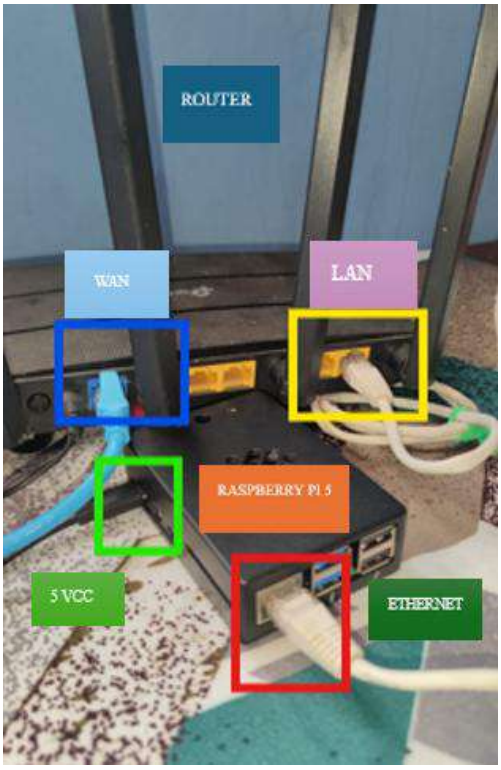
- Monitorear el trafico de red
- Verificar la configuracion de Suricata IDS
- No tomar acciones urgentes

3.6.7 Implementación de Hardware

El hardware del sistema se instaló en un área central del hogar, segura y ventilada, para optimizar la cobertura inalámbrica. Los dispositivos se ubicaron juntos en una repisa, permitiendo su correcta interconexión y funcionamiento. En la figura 6, se puede apreciar de manera más sencilla la conexión de los equipos a través de un cable ethernet.

Figura 55

Conexión posterior de los equipos de red



3.6.7.1 Núcleo de Procesamiento y Memoria (Raspberry Pi 5)

El hardware principal del sistema se basó en la Raspberry Pi 5, seleccionada por su arquitectura de procesador Broadcom BCM2712 Quad-core Arm Cortex-A76 a 2.4 GHz de 64 bits. Esta capacidad de cómputo es crítica para la viabilidad del proyecto, ya que el sistema debe ejecutar de forma simultánea el análisis de inspección profunda de paquetes de Suricata, y los contenedores Docker de Pi-hole.

Para evitar problemas con los arranques de servicios, la placa cuenta con 8 GB de memoria RAM LPDDR4X, lo que garantiza un flujo de datos continuo y sin latencia perceptible para los usuarios de la red local.

3.6.7.2 Interfaces de Comunicación de Red

La topología física requirió el uso intensivo de dos interfaces de red separadas lógicamente. Para la conexión de área amplia WAN, se utilizó el puerto Gigabit Ethernet integrado en la SBC, el cual se conectó mediante cableado UTP Categoría 5e al enrutador del proveedor de servicios de interney. Para la red de área local LAN, se habilitó el módulo transceptor inalámbrico integrado, configurado para emitir la señal Wi-Fi en la banda de 2.4 GHz.

Esta separación física de las interfaces es lo que permite que la Raspberry Pi actúe como un nodo intermediario de filtrado obligatorio entre los dispositivos móviles y el acceso a Internet.

3.6.7.3 Almacenamiento

Observando que el sistema tiene que almacenar las bases de datos MariaDB locales, la configuración del servidor y un volumen significativo de logs o registros que genera el Sistema de Detección de Intrusos se optó por usar una unidad de almacenamiento microSD SDR104 de 128 GB.

3.6.7.4 Gestión Energética y Disipación Térmica

Dada la condición del control parental que ha de servir 24/7, dado que ha de estar procesando tráfico en tiempo real, la gestión energética y térmica resultó ser una condición de diseño necesaria. La placa fue alimentada mediante una fuente de tensión continua de 5.1 voltios capaz de suministrar la corriente necesaria para operar bajo altas cargas.

Por otro lado, dado que el procesador de la Raspberry Pi 5 tiende a elevar su temperatura al ejecutar servicios y sistemas desde su inicio de encendido, para lo cual se incorporó un sistema de refrigeración con un disipador y ventilador, acompañado por una pasta térmica en forma de almohadillas, para prevenir el calor térmico, asegurando así un rendimiento óptimo y prolongando la vida útil del hardware.

Tabla 18

Especificación de interfaces y puertos

Dispositivo	Puerto / Interfaz	Función en el prototipo
Raspberry Pi 5	Ethernet (eth0)	Salida a Internet. Envía al router ISP el tráfico ya analizado y filtrado.
Raspberry Pi 5	Wi-Fi (wlan0)	Configurado como Punto de Acceso 2.4 GHz. Conecta los dispositivos de la red doméstica.
Raspberry Pi 5	USB-C (5.1V)	Alimentación eléctrica (5.1V / 5A) para soportar el consumo de la Pi 5.
Router ISP	LAN (Ethernet)	Recibe el cable desde la Raspberry y permite la salida a Internet.
Router ISP	WAN	Conecta el router con la red del proveedor (acceso a Internet).

Capítulo IV: Pruebas de Funcionamiento

Este capítulo muestra las pruebas que se han realizado para evaluar el funcionamiento del sistema, tanto en su parte hardware como software. El propósito consiste en comprobar de manera objetiva que cada uno de los módulos llega a cumplir con los requerimientos y también identificar las limitaciones del prototipo, así como su rendimiento en distintas situaciones de uso. Los resultados se muestran de manera clara, transparente y basada en las evidencias obtenidas durante las pruebas, permitiendo verificar el correcto funcionamiento del sistema y señalar los aspectos que podrían mejorarse en futuras versiones.

4.1 Plan de Pruebas

Con el fin de verificar el funcionamiento del prototipo de control parental, se elaboró un plan de pruebas que organiza las evaluaciones realizadas sobre los principales módulos del sistema. Gracias a este plan fue posible realizar la verificación del cumplimiento de los requisitos funcionales y la evaluación del comportamiento del prototipo bajo diferentes condiciones de operación.

Las pruebas se clasifican en funcionales, rendimiento y estrés. Las pruebas funcionales verificaron el correcto funcionamiento la API REST, Pi-hole, la heurística, Suricata, la aplicación cliente ChildGuard, las notificaciones y la generación de reportes. Las pruebas de rendimiento y estrés, en cambio, evaluaron el comportamiento del sistema al aumentar el número de dispositivos conectados, evaluando el uso de recursos de la Raspberry Pi 5 y la estabilidad de los servicios implementados.

Tabla 19

Nomenclatura del plan de pruebas

ID	Descripción
PS-CL	Software: Prueba del módulo de conexión lógica
PS-R	Software: Prueba del módulo de red
PS-G	Software: Prueba del módulo de gestión
PS-MI	Software: Prueba del módulo de Monitoreo e Interfaz
PS-AH	Software: Prueba del módulo de Análisis Heurístico
PH-IR	Hardware: Prueba de Interfaz de red
PH-RO	Hardware: Prueba de Rendimiento óptimo

En la sección de pruebas se lleva a cabo un numero de pruebas para determinar el buen funcionamiento del sistema de control parental, paro lo cual se debe realizar un plan de pruebas.

Tabla 20

Resumen de plan de pruebas

ID	Prueba	Tipo	Prioridad	Estado
PS-CL01	Conectividad API REST	Funcional	Alta	Realizado
PS-R01	Red hostpod	Rendimiento	Alta	Realizado
PS-R02	Autenticación	Funcional	Alta	Realizado
PS-G01	Bloqueo/desbloqueo web	Funcional	Alta	Realizado
PS-G02	Corte de internet	Funcional	Alta	Realizado

PS-MI01	Monitoreo ChildGuard	Funcional	Alta	Realizado
PS-MI02	Alertas y notificaciones	Funcional	Alta	Realizado
PS-AH01	Análisis heurístico y reporte general	Funcional	Alta	Realizado
PH-IR01	Cobertura de la señal	Rendimiento	Media	Realizado
PH-RO01	Rendimiento del SBC	Rendimiento	Alta	Realizado

Este paso se lo desarrolla para revisar y analizar si el sistema tuvo algún problema con los requisitos planteados en un comienzo, si es así pues se toma en cuenta para colocarlo como una limitación dentro del sistema planteado. Con ello se corrobora que el sistema funciona de manera adecuada cuando tiene alguna dificultad.

4.1.1 PS-CL01: Pruebas de Conexión Lógica

Tabla 21

Conectividad y disponibilidad de la API REST

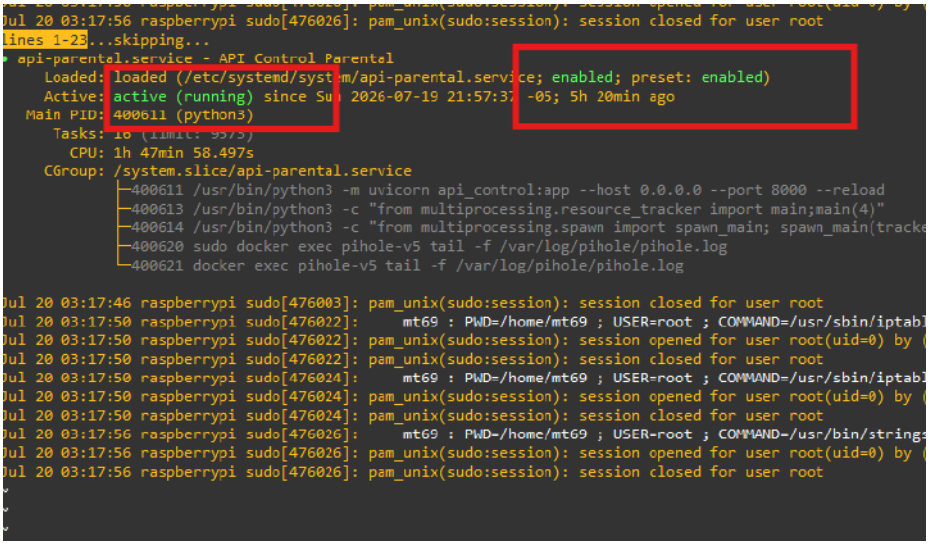
Campo	Detalle
Objetivo	Verificar que el backend FastAPI responde correctamente
Precondición	Raspberry Pi 5 encendida, servicio api-parental.service activo
Comando	curl -s http://192.168.10.1:8000/dispositivos/activos
Resultado esperado	HTTP 200, lista de dispositivos en JSON

Resultado obtenido	HTTP 200: 1 dispositivo activo detectado

En esta prueba se va a validar que el backend que se desarrolló en FastAPI responde correctamente a las solicitudes HTTP realizadas desde la aplicación del administrador. Antes que nada, primero se debe revisar si el servicio que en este caso es la API esta activo y corriendo sin ningún problema.

Figura 56

Servicio de API activo



Luego se puede realizar una consulta desde otro dispositivo para verificar que la API esté funcionando correctamente y que responde a solicitudes HTTP.

Figura 57

Verificación de API

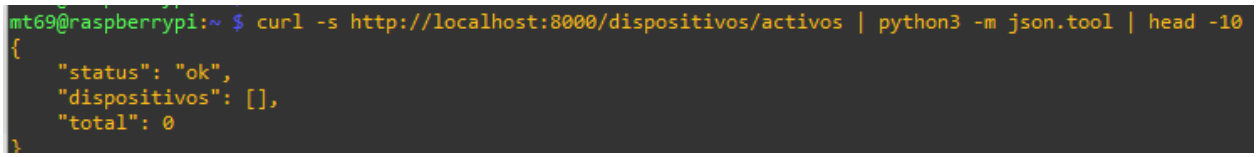


Se realizó una solicitud HTTP GET a la documentación interactiva de la API (/docs) utilizando la dirección IP de la Raspberry Pi 5 y el puerto 8000 desde un equipo conectado a la misma red.

También se puede realizar otro tipo de consultas para seguir verificando que el API está en buen estado, en este caso se hace una consulta a los cuantos dispositivos hay activos.

Figura 58

Dispositivos activos en la red



Resultado obtenido

El servidor respondió con el código HTTP 200 OK, confirmando el correcto funcionamiento del backend desarrollado con FastAPI. Además, fue posible acceder a la documentación interactiva de la API, donde se verificó la disponibilidad de los endpoints implementados.

4.1.2 PS-R01: Prueba de Hostpod

Tabla 22

Capacidad de monitoreo de dispositivos

Campo	Detalle
Objetivo	Verificar que el sistema pueda monitorear simultáneamente varios dispositivos Android sin pérdida de comunicación.
Precondición	Raspberry Pi 5 en funcionamiento, API activa y dispositivos vinculados mediante ChildGuard.
Procedimiento	Establecer conexión secuencial de múltiples dispositivos y verificar que la IP sea asignada a cada uno de ellos
Resultado esperado	Todos los dispositivos deben estar conectados y acomodar la nueva información continuando con el funcionamiento normal, si se tratase de un fallo de sincronización.

Lo primero que se realiza es conectar un dispositivo de prueba como una computadora o teléfono móvil a la red WiFi creada por la Raspberry Pi 5 esta tendrá la IP 192.168.10.1 que será el Gateway.

Figura 59

Asignación de dirección IP

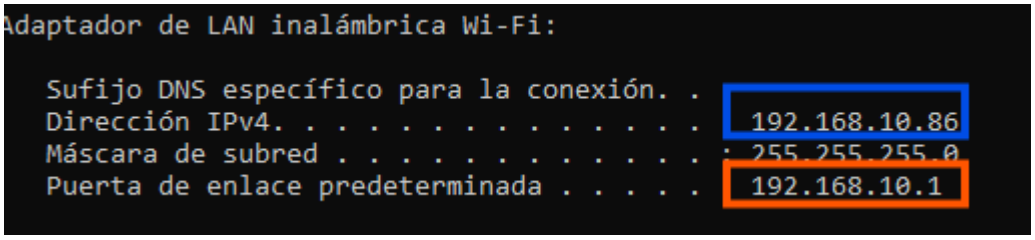


Figura 60

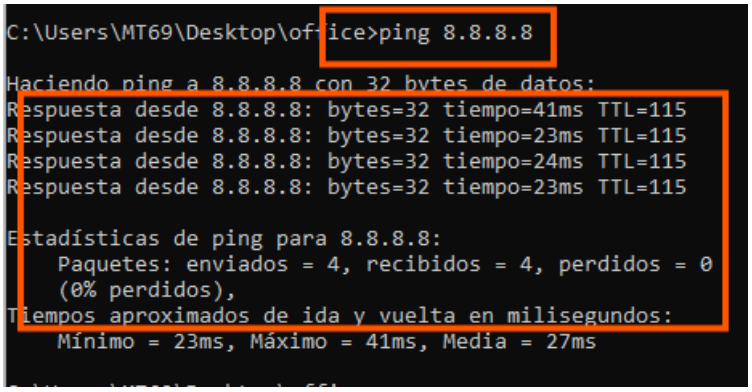
Pantalla de dispositivos conectados



En la pantalla de la sección de dispositivos podemos observar que hay dos dispositivos conectados en este caso el que dice MT (Laptop), es el dispositivo donde se va a realizar las pruebas. Desde el dispositivo, se ejecuta una prueba de conectividad continua hacia Internet mediante el comando ping 8.8.8.8., con ello se corrobora que tenga acceso a internet sin ningún inconveniente.

Figura 61

Prueba de conectividad



4.1.3 PS-R02: Prueba de autenticación de usuario

Tabla 23

Pruebas de autenticación

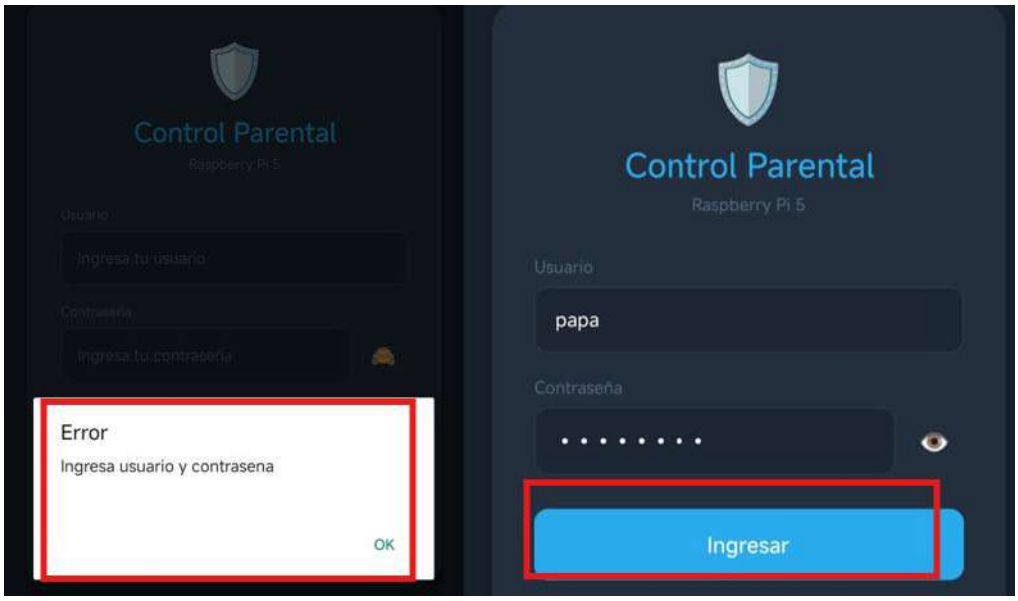
Campo	**Detalle**
Objetivo	Verificar el funcionamiento del proceso de autenticación para el acceso a la aplicación administrador.
Precondición	API REST funcionado y credenciales existentes en la base de datos.
Procedimiento	Ingresar credenciales válidas e inválidas desde la aplicación administrador y verificar la respuesta del sistema.
Resultado esperado	El sistema debe permitir el acceso únicamente con credenciales válidas e impedir los intentos de acceso con información errónea.

Inicialmente, se verificó el proceso de autenticación de usuarios mediante el ingreso de credenciales en el módulo de inicio de sesión de la aplicación del administrador. Para acceder al

sistema, el usuario debe ingresar correctamente su nombre de usuario y contraseña, siguiendo un proceso similar al utilizado en otras plataformas.

Figura 62

Verificación del administrador



Durante la prueba, se comprobó que, cuando las credenciales ingresadas coinciden con los datos registrados en la base de datos MariaDB, el sistema permite el acceso al panel principal o dashboard del administrador. En caso contrario, se genera un mensaje de error indicando que las credenciales son incorrectas y solicitando al usuario volver a ingresar los datos de acceso.

Resultado obtenido:

El ingreso ha sido exitoso de manera correcta conectándose tanto al servidor local como de manera remota con ayuda la Tailscale, con ello se verifica que la sección de Login está de manera correcta sin errores.

4.1.4 PS-G01: Prueba de bloqueo/desbloqueo de sitios web

Para realizar las pruebas tanto de bloqueo como de desbloqueo de la las páginas web, se hace uso del servicio de DNS en este caso con Pi-hole. Al momento de ingresar a la sección de Redes, en esta vamos a realizar un bloqueo a una página de red social en este caso “Facebook”, no es necesario colocar todos los dominios, basta con colocar la palabra de la página.

Tabla 24

Bloqueo y desbloqueo web

Campo	**Detalle**
Objetivo	Verificar el bloqueo y desbloqueo de sitios web mediante Pi-hole desde la aplicación administrador.
Precondición	Pi-hole operativo y dispositivo conectado utilizando la Raspberry Pi como servidor DNS.
Procedimiento	Agregar un dominio a la lista de bloqueo y posteriormente eliminarlo para comprobar el cambio en el acceso al sitio web. Todo mediante la APP.
Resultado esperado	El dominio debe bloquearse al ser agregado a la lista negra y volver a estar disponible después de eliminar la restricción.

Figura 63

Bloqueo de la página web mediante la aplicación



En este paso tendremos ayuda de IA, en el backend está diseñado un pront que nos ayuda a buscar todos los dominios posibles en ese momento, cabe recalcar que los dominios pueden aumentar y la IA está preparada para actualizar la lista.

Figura 64

Página web bloqueado y agregado a la lista negra



Figura 65

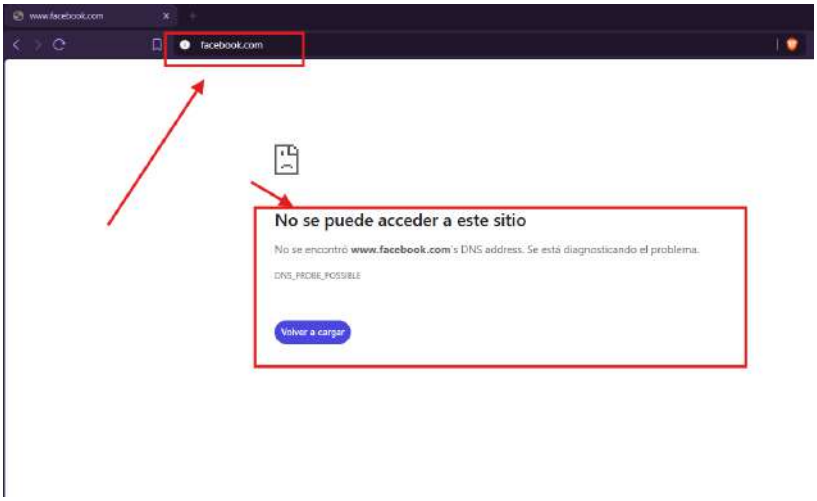
Registro de huellas digitales de Facebook.



Para corroborar esto hay que revisar si en el dispositivo que está siendo monitoreado y administrado, se ha realizado este cambio, en este caso se hace uso de un computador para las pruebas. Se hace la inspección de la red social y como se puede verificar la página web está bloqueada por el sistema.

Figura 66

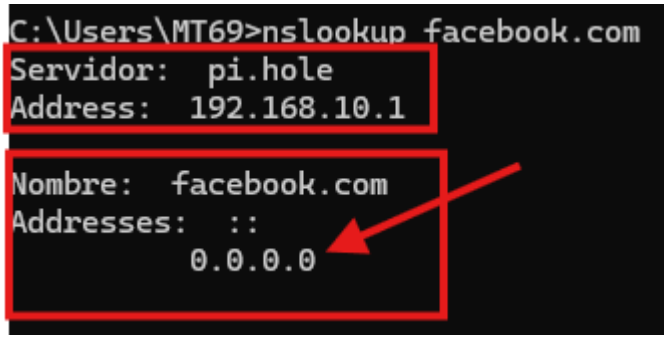
Página de red social bloqueada



Para comprobar que las solicitudes DNS están siendo gestionadas correctamente por Pi-hole, se realiza una consulta mediante el comando nslookup facebook.com. Con esta prueba se verifica si el dominio está siendo resuelto normalmente o si ha sido bloqueado por el sistema de filtrado.

Figura 67

Verificación de resolución DNS



Una vez realizada la prueba de bloqueo del sitio web, se procede a verificar el funcionamiento del proceso de desbloqueo. Para ello, únicamente se elimina el dominio de la lista de bloqueo configurada en el sistema. Después de esta acción, las reglas asociadas al bloqueo se actualizan automáticamente, permitiendo nuevamente el acceso al sitio web de manera correcta.

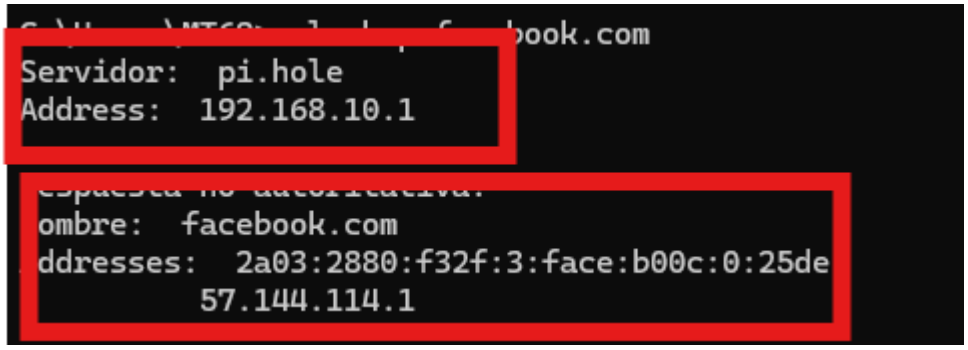
Figura 68

Desbloqueo de la página web



Figura 69

Resolución DNS autorizada por Pi-hole



Durante las pruebas, el sistema permitió bloquear y desbloquear los sitios correctamente, sin presentar inconvenientes. Se observó que, al volver a bloquear una página después de haberla desbloqueado, algunos dispositivos continúan accediendo al contenido durante uno o tres minutos debido a la memoria caché del navegador o del sistema DNS. Transcurrido ese tiempo, el bloqueo se aplica correctamente y el acceso al sitio queda restringido.

Resultado obtenido

El acceso fue bloqueado exitosamente. Pi-hole redirigió la consulta DNS a 0.0.0.0 impidiendo la carga del sitio. El sistema registró la alerta correspondiente y envió una notificación al dispositivo del padre mediante la integración de OneSignal.

4.1.5 PS-G02: Prueba de corte y horarios de internet

Tabla 25

Corte de Internet

Campo	Detalle
Objetivo	Verificar el funcionamiento del corte y restablecimiento del acceso a Internet desde la aplicación administrador.
Precondición	Dispositivo cliente conectado a la red administrada por la Raspberry Pi 5.
Procedimiento	Activar la opción de corte de Internet desde la aplicación administrador y posteriormente restablecer la conexión.
Resultado esperado	El dispositivo debe perder el acceso a Internet durante el bloqueo y recuperarlo al restablecer el servicio.

Esta prueba se realiza mediante la aplicación del administrador, específicamente en las secciones de horarios y corte de Internet. Ambas funcionalidades trabajan bajo el mismo mecanismo de bloqueo y desbloqueo del acceso a la red; sin embargo, se diferencian en que la configuración de horarios depende de la hora establecida por el administrador.

Para la función de horarios, el sistema debe validar la hora programada antes de aplicar la restricción, por lo que puede existir un tiempo de espera aproximado entre 30 segundos a 1 minuto para ejecutar el bloqueo o restablecer nuevamente la conexión a Internet.

Figura 70

Horarios de internet

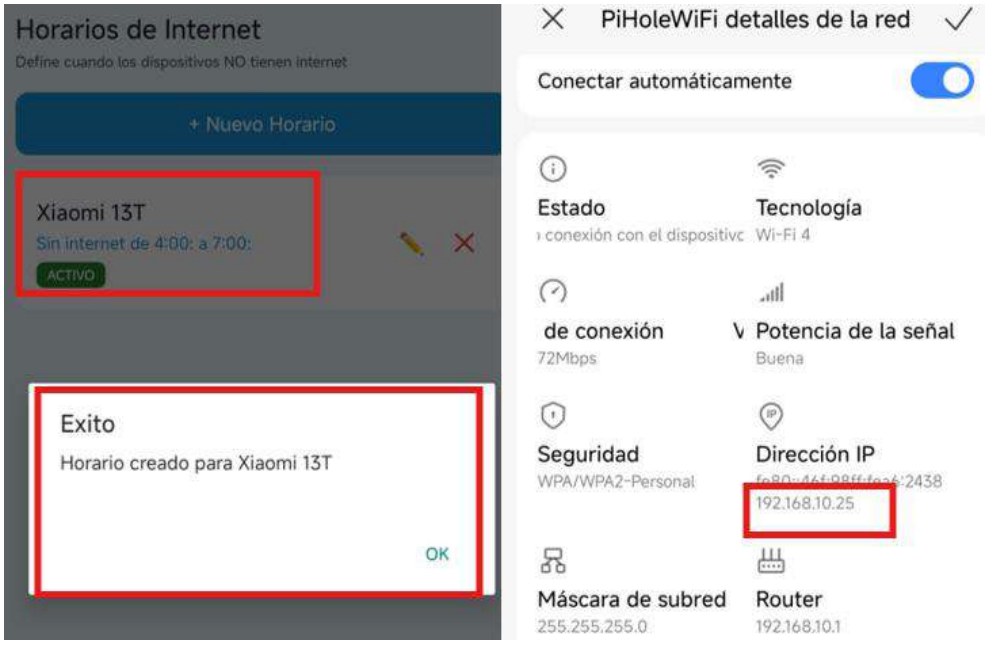
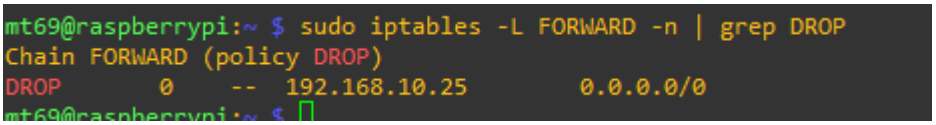


Figura 71

Dispositivo bloqueado por horarios



De manera similar, la función de corte de Internet, opera mediante un mecanismo de bloqueo implementado en el backend del sistema. A diferencia de los horarios programados, esta opción ejecuta el bloqueo o restablecimiento de la conexión de forma inmediata, permitiendo al administrador controlar el acceso a Internet del dispositivo en tiempo real.

Luego mediante la aplicación el administrador puede realizar un corte de internet a ese dispositivo conectado de esa manera hará que no tenga acceso a internet. Y cuando lo haga va a haber una opción para reconectarlo nuevamente.

Figura 72

Cortar el internet mediante app

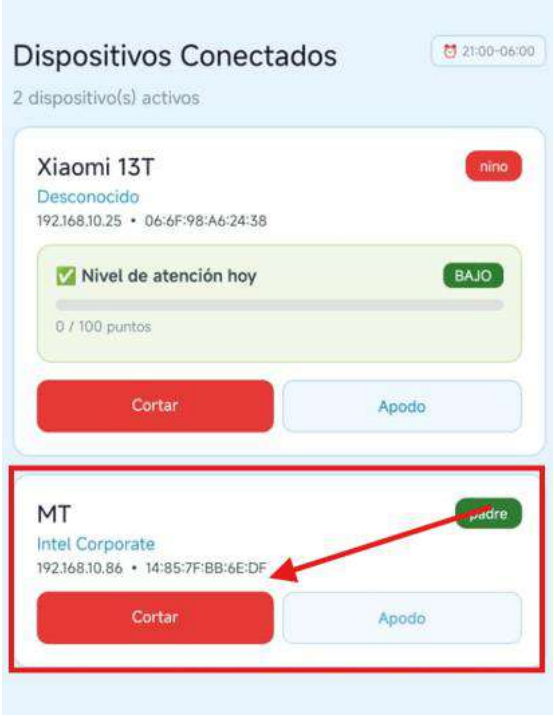
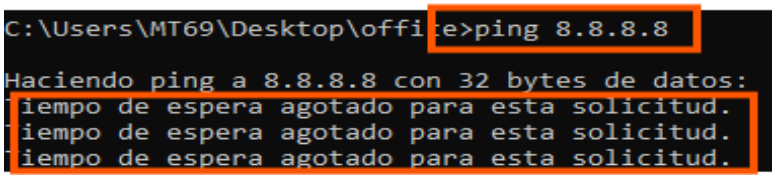


Figura 73

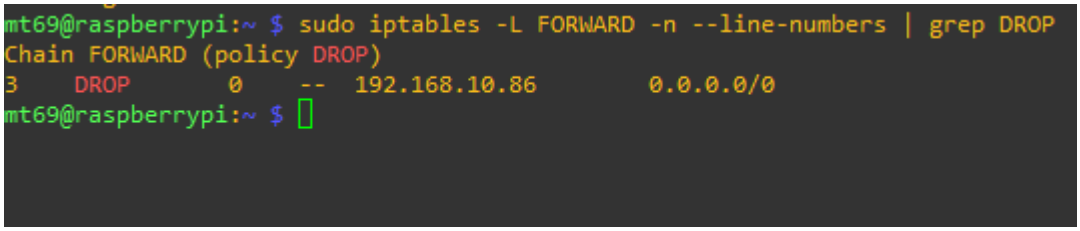
Corte de internet a cliente



Para revisar si es proceso salió según lo planificado, se puede realizar una consulta directa a la Raspberry Pi 5 mediante el comando sudo iptables -L FORWARD -n –line-numbers | grep DROP. Con eso confirmamos que la regla se aplicó correctamente.

Figura 74

Creación de regla para denegar acceso a internet



Y para restaurar el internet basta con realizar la activación en el aplicativo, se lo puede realizar de forma individual o si hay más dispositivos bloqueados se los puede realizar de forma general la activación del internet. Toda esta prueba se lo hace en tiempo real, es decir al momento de cortar o activar el internet se lo hace de manera inmediata con un retraso de 0.01 segundos.

Figura 75

Restauración del internet



Resultado obtenido

El sistema realizó correctamente el bloqueo y la restauración del acceso a Internet sin afectar el funcionamiento de los demás servicios con ello se finaliza las pruebas de corte de internet a cada dispositivo.

4.1.6 PS-MI01: Prueba de Gestión de Dispositivos

En esta parte se comprueba la supervisión de los dispositivos conectados gracias al ChildGuard. Una vez que un dispositivo Android ha sido conectado correctamente a la plataforma, el administrador lo puede gestionar mediante la aplicación del control parental y utilizar las diferentes funciones de supervisión.

Entre las funcionalidades más importantes se encuentran las opciones de bloqueo y desbloqueo de aplicaciones, la revisión de las aplicaciones que más tiempo se utilizan, la consulta del estado de carga de la batería, la comprobación del historial de las aplicaciones que se han instalado y más opciones para la gestión del dispositivo.

Tabla 26

Monitoreo ChildGuard

Campo	**Detalle**
Objetivo	Asegurarse que la información del dispositivo cliente se envía al servidor, el cual mantiene la funcionalidad de la aplicación ChildGuard.
Precondición	La aplicación ChildGuard se encuentra instalada y vinculada a la aplicación administrador.
Procedimiento	Ejecutar la aplicación cliente y verificar la actualización de información como batería, aplicaciones instaladas, tiempo de uso y estado del dispositivo.

Resultado esperado	La información del dispositivo debe estar correctamente sincronizada y, por tanto, visualizada desde la aplicación administrador.

Es importante que el dispositivo se encuentre previamente vinculado al sistema, ya que este proceso permite establecer la comunicación entre la aplicación de administración y ChildGuard.

Figura 76

Código de vinculación hacia el control parental



Figura 77

Dispositivo vinculado correctamente

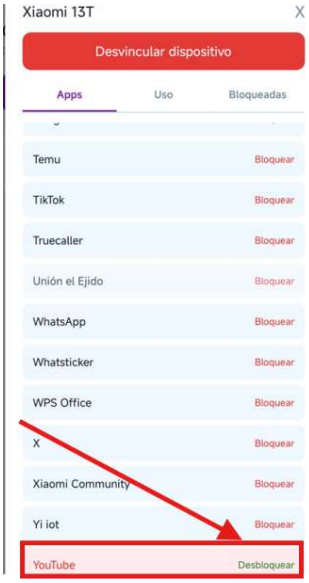


Bloqueo de aplicaciones

Para comprobar el correcto funcionamiento de estas funciones, se realiza una prueba de bloqueo de una aplicación instalada en el dispositivo.

Figura 78

Bloqueo de aplicación desde la app de administrador



Una vez aplicado el bloqueo, la aplicación seleccionada ya no puede utilizarse en el dispositivo. Cuando el usuario intenta abrirla, esta se cierra automáticamente, impidiendo su ejecución. Además, esta acción queda registrada en el sistema y se incorpora al reporte general, por lo que puede visualizarse posteriormente al generar el informe en formato PDF. Pero una vez desbloqueado se vuelve a usar normalmente.

Figura 79

Verificación dentro del reporte general



Resultado obtenido

La aplicación no pudo ejecutarse al intentar abrirla. El evento fue registrado correctamente y quedó disponible en el reporte general del sistema.

4.1.7 PS-MI02: Prueba de alertas y notificaciones

Tabla 27

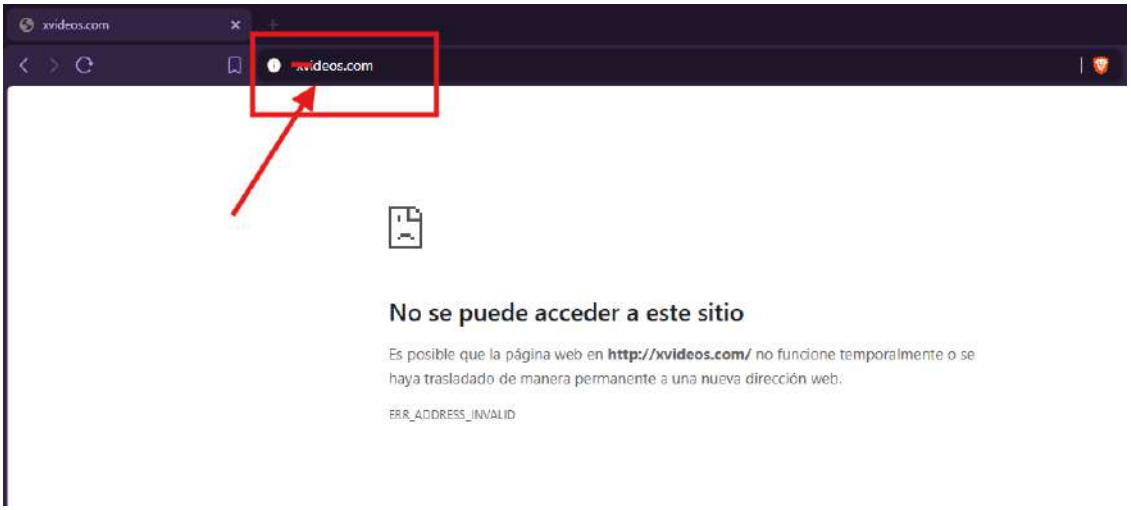
Prueba de alertas y notificaciones

Campo	Detalle
Objetivo	Verificar la generación y envío de alertas cuando se detecta un evento de riesgo.
Precondición	Sistema de monitoreo activo, OneSignal configurado y dispositivo registrado.
Procedimiento	Acceder a un dominio clasificado como restringido y verificar la generación de la alerta y la recepción de la notificación.
Resultado esperado	El sistema debe registrar la alerta y enviar una notificación al administrador en tiempo oportuno.

Para comprobar que el sistema de alertas funciona, se visita una página web que se encuentra en la lista negra, en este caso una página dedicada a adulto. Esta clase de páginas son las que tienen un nivel más alto de riesgo para el sistema, por lo que en el momento en el que se intenta acceder a la misma se genera la alerta inmediatamente. En cambio, si se intenta acceder a una página de Facebook, aunque esté bloqueada, su nivel de riesgo es menor, y por lo tanto el peso que tendrá dentro del algoritmo será menos importante y la alerta no se disparará con la misma prioridad.

Figura 80

Intento de ingreso a página de lista negra y prohibida



La notificación se genera de forma inmediata, ya que el sistema identifica que el usuario intentó acceder a un sitio web con contenido prohibido. En ese momento, se envía una alerta al padre o tutor, permitiéndole conocer en tiempo real la actividad que está realizando el menor y tomar las medidas que considere necesarias.

Figura 81

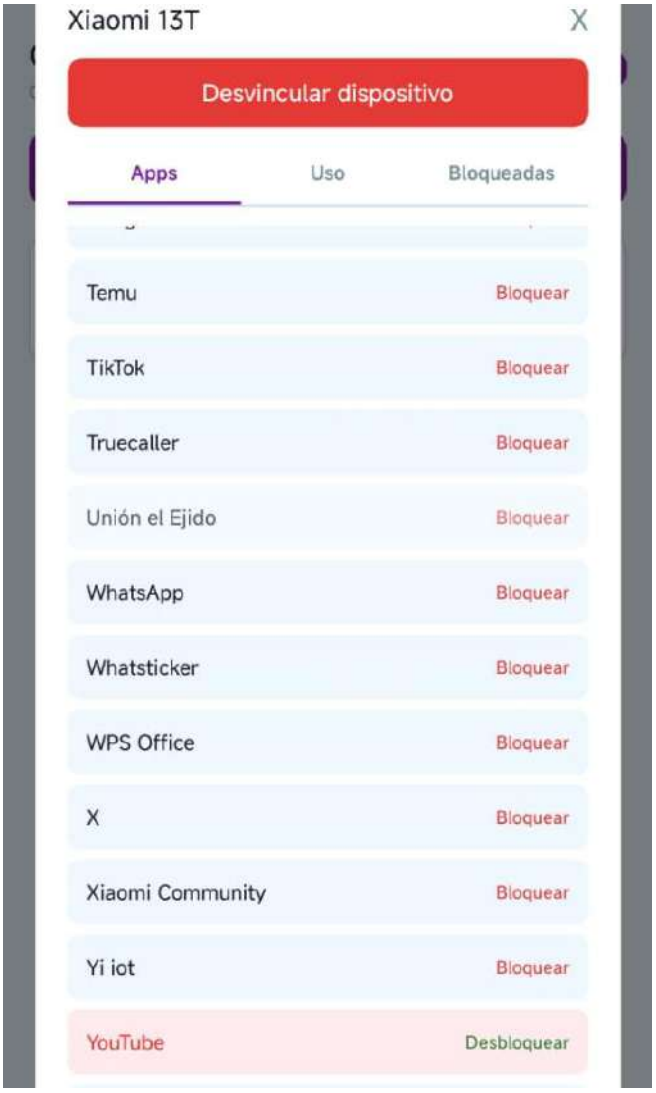
Notificación de intento a sitios prohibidos



Para otro tipo de alertas por ejemplo cuando desde la app del administrador se realiza un bloqueo de una aplicación hacia el dispositivo que está monitorizando, en este caso se realiza un bloqueo a la app de “Youtube”, esta se bloqueara en un lapso de 15 segundos.

Figura 82

Bloqueo de aplicaciones

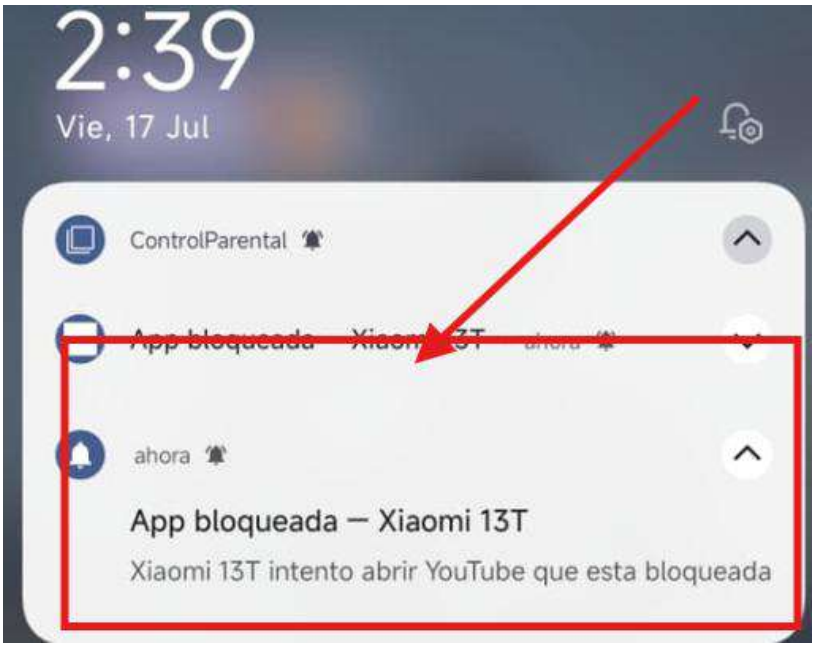


Cuando se ejecuta esta acción, el evento queda registrado en el sistema. Si el dispositivo intenta acceder a la aplicación bloqueada, el acceso será denegado mediante la aplicación

ChildGuard, la cual se encuentra vinculada con la aplicación del administrador. Además, el administrador recibirá una notificación de alerta informando sobre el intento de acceso a una aplicación que está bloqueada.

Figura 83

Notificación de app bloqueada



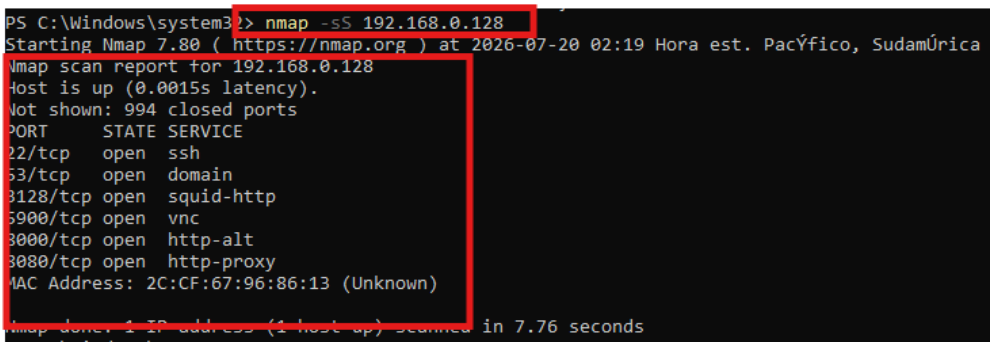
Para cada tipo de evento generado por el sistema se dispone de una alerta o notificación correspondiente. En el caso de posibles amenazas dentro de la red, el sistema cuenta con un IDS que supervisa constantemente el tráfico, siendo Suricata el encargado de detectar e informar sobre actividades sospechosas.

Para validar su funcionamiento, se realizó una prueba de detección mediante un escaneo de puertos ejecutado desde un computador utilizando el comando correspondiente, con el objetivo de verificar la generación de alertas ante comportamientos anómalos en la red.

nmap -sS 192.168.0.128

Figura 84

Prueba de ataque a puertos



Cuando se realiza esta acción casi de manera inmediata llega una notificación sobre un posible ataque a los puertos de la Pi 5 que se encuentran vulnerables. En esa cuestión hace dos cosas el sistema, la una manda la notificación y la otra es la alerta.

Figura 85

Notificación de ataque a puertos

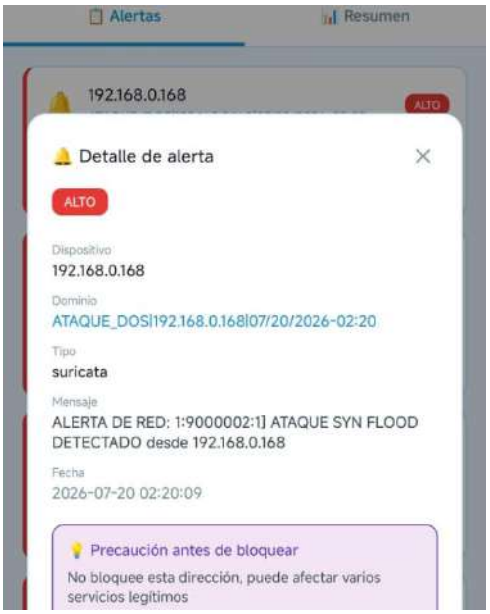


La alerta generada por Suricata se visualiza dentro de la aplicación del administrador, donde se integra un asistente basado en inteligencia artificial que permite interactuar y obtener información sobre el evento detectado. Este asistente ayuda al padre o administrador a comprender qué está ocurriendo y qué acciones puede tomar de manera sencilla. En este caso,

una de las acciones disponibles es el bloqueo de la dirección IP de origen asociada a la actividad sospechosa

Figura 86

Alerta de ataque de puertos



Con ello podrá realizar alguna de las acciones que da la IA, en este caso solo podría bloquear esa IP de originen. Si es el caso que bloqueen una IP, esta se presenta en la sección de dispositivos en una lista negra de IP bloqueadas.

Resultado obtenido

El sistema de notificaciones obtuvo resultados satisfactorios gracias a la integración del servicio de OneSignal, el cual permitió el envío de notificaciones en tiempo real. De esta manera, el administrador recibe las alertas generadas por el sistema incluso cuando la aplicación se encuentra en segundo plano o el dispositivo móvil está bloqueado, garantizando una comunicación oportuna ante cualquier evento detectado.

4.1.8 PS-AH01: Pruebas del análisis heurístico

Tabla 28

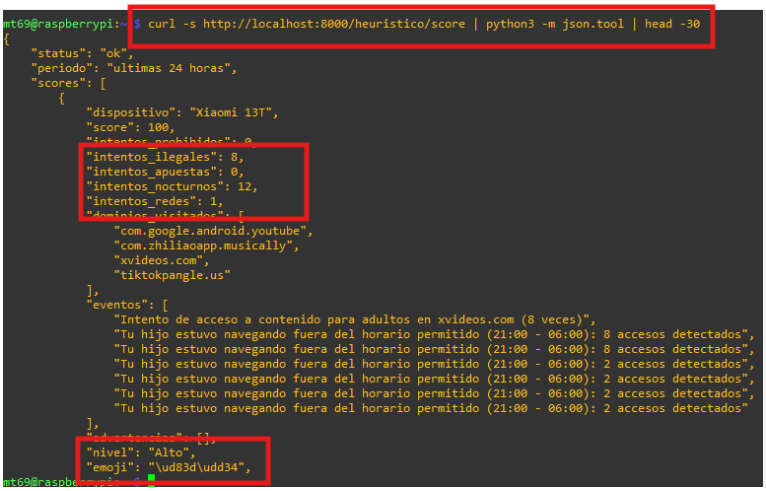
Análisis heurístico y reporte general

Campo	Detalle
Objetivo	Verificar el funcionamiento del módulo de análisis heurístico y la generación del reporte general del comportamiento del dispositivo.
Precondición	Sistema de monitoreo activo, registros DNS disponibles y módulo heurístico en ejecución.
Procedimiento	Generar actividad de navegación con diferentes categorías de sitios web y solicitar la creación del reporte desde la aplicación administrador.
Resultado esperado	El sistema debe calcular el nivel de riesgo de acuerdo con los eventos registrados y generar un reporte con el resumen de la actividad detectada.

En este caso se va a realizar evaluación que el sistema analiza correctamente el comportamiento de navegación de los dispositivos conectados. La prueba se lleva a cabo dentro del backend es decir se va a realizar una consulta al endpint correspondiente.

Figura 87

Consulta al Score del análisis heurístico



Además, se realizó una prueba mediante un escaneo de puertos controlado para verificar el funcionamiento de Suricata IDS. Como resultado, el sistema detectó la actividad sospechosa, generó una alerta y envió una notificación al administrador en tiempo real. Asimismo, el módulo de inteligencia artificial analizó el evento, identificó el tipo de ataque y proporcionó información sobre la dirección IP de origen, facilitando la interpretación del incidente y la toma de decisiones.

Resultado obtenido

El sistema ha podido llevar a cabo el cálculo del score de riesgo hace siempre correcto utilizando la metodología de análisis de las categorías de navegación, los patrones de uso, los horarios de entrada y la clasificación inteligente de los dominios junto a la obtención de recomendaciones gracias al modelo de IA embebido. Suricata detectó correctamente el escaneo de puertos, registró el evento y el sistema notificó al administrador sobre la posible amenaza detectada.

Como etapa final del análisis, el sistema permite generar reportes diarios que recopilan toda la información obtenida durante el monitoreo. Estos reportes incluyen datos de los

dispositivos registrados, el estado de la red, el comportamiento de navegación, las alertas detectadas y demás eventos relevantes del sistema.

Para poder comprobar que la funcionalidad de la aplicación funcionaba correctamente se llevó a cabo la generación del correspondiente reporte desde la aplicación del administrador y a continuación se descargó el correspondiente archivo PDF que se había generado. Finalmente se procedió a comprobar que el archivo en sí era correcto, es decir, que el documento fuera creado correctamente incluyendo la información recogida por el mismo y la correspondiente y necesaria información que había sido generada por la IA en el análisis.

Figura 88

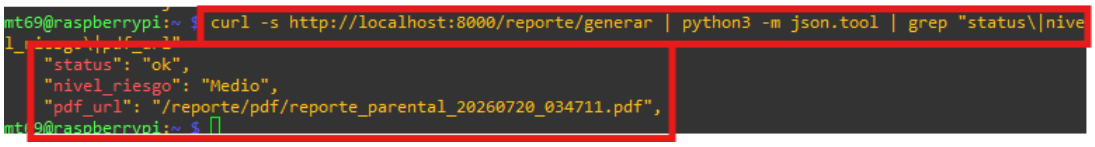
Generación de reporte mediante la Aplicación de administrador



La generación de reportes también puede llevarse a cabo de manera manual desde la consola, lo que posibilita incluso poder verificar el correcto funcionamiento del procedimiento antes de que este pase a integrarse con el frontend. Este método fue utilizado durante la etapa de desarrollo para realizar diferentes pruebas y validar la información generada por el sistema.

Figura 89

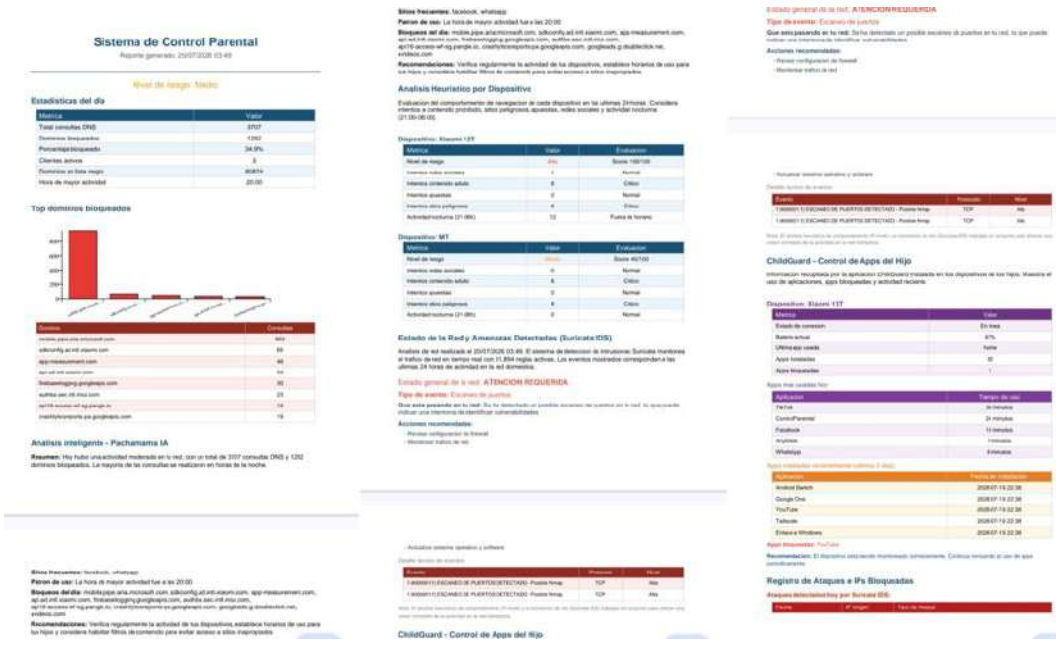
Generación de PDF por consola



Una vez generado el archivo, es posible visualizar el reporte en formato PDF con un resumen detallado de la actividad registrada durante el día. Entre la información presentada se incluyen las páginas web más visitadas, las horas de mayor actividad, el análisis heurístico del comportamiento de navegación, el estado de la red, las alertas detectadas y la información recopilada de cada dispositivo mediante la aplicación ChildGuard, proporcionando una visión integral del funcionamiento del sistema. En el anexo 3 se encuentra mas detallado el reporte generado.

Figura 90

PDF con los registro y reportes del día



Resultado obtenido:

El reporte fue generado correctamente en formato PDF, incorporando las estadísticas del sistema, el análisis heurístico, los eventos de seguridad, las recomendaciones de la inteligencia artificial y el historial de actividades registradas.

4.1.9 PH-IR01: Cobertura de la señal

Tabla 29

Prueba de Cobertura de la señal

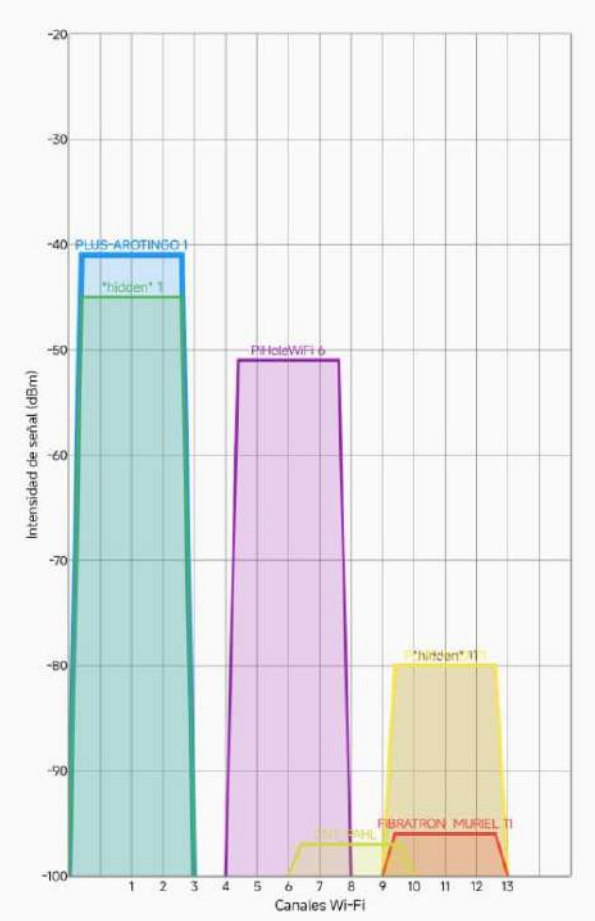
Campo	**Detalle**
Objetivo	Verificar que la señal WiFi generada por la Raspberry Pi 5 a través de la interfaz wlan0 tiene cobertura suficiente para que los dispositivos de la red doméstica se conecten.
Precondición	Raspberry Pi 5 encendida con hostapd activo, interfaz wlan0 configurada como punto de acceso con SSID PiHoleWiFi.
Procedimiento	Conectar el dispositivo Xiaomi 13T a la red PiHoleWiFi desde diferentes ubicaciones del hogar. Medir la intensidad de señal (dBm).
Resultado esperado	El dispositivo debe conectarse exitosamente a la red WiFi generada por la Raspberry Pi 5 hasta los 10 metros debe ser optima la señal.

La prueba tuvo como objetivo verificar que la red inalámbrica generada por la Raspberry Pi 5 mediante la interfaz wlan0 proporcionara una cobertura suficiente para los dispositivos de la red doméstica, garantizando una conexión estable y el funcionamiento correcto del sistema de control parental. En este caso se hizo uso de una aplicación denominada WifiAnalyzer con la cual

se pudo medir la intensidad de la señal en diferentes partes de la casa además que se puede observar algunos parámetros adicionales como el canal en el que se encuentra.

Figura 91

Uso de Wifianalyzer dentro de la red domestica



Las pruebas se realizaron en varias partes de la casa y fuera de ella, en primera instancia se lo realizo junto a la SBC, es decir a un metro de distancia, dando como resultado -44dBm, desde ahí se va a partir como referencia de una señal excelente.

Figura 92

Intensidad de la señal a 1 metro de distancia



En la sala, ubicada a aproximadamente cinco metros del punto de acceso, se obtuvo una intensidad de -59 dBm, correspondiente a una señal muy buena. Durante la prueba el dispositivo mantuvo una conexión estable, recibió correctamente una dirección IP asignada por dnsmasq y todas las consultas que realiza al servidor son buenas.

Figura 93

Intensidad mediada a 5m de distancia



A una distancia de ocho metros que se encuentra la cocina, se registró una intensidad de señal de -66 dBm, considerada una señal buena para aplicaciones de navegación y comunicación en tiempo real. El dispositivo permaneció conectado sin interrupciones y el acceso a Internet se realizó con normalidad.

Figura 94

Intensidad mediada a 8m de distancia



En los dormitorios, situados aproximadamente a 10 metros de la Raspberry Pi 5, la intensidad de señal fue de -69 dBm. Aunque se observó una disminución del nivel de potencia debido a la mayor distancia y a la presencia de obstáculos propios de la vivienda, la conexión permaneció estable

Figura 95

Intensidad mediada a 10m de distancia



En el patio, a una distancia aproximada de 13 metros, se obtuvo una intensidad de señal de -85 dBm, valor considerado aceptable para mantener la conectividad. A pesar de tratarse del punto más alejado del acceso inalámbrico, el Xiaomi 13T logró permanecer conectado a la red PiHoleWifi, pero ya muy limitado para las diferentes actividades.

Figura 96

Intensidad mediada a 13m de distancia



Tabla 30

Sitos de pruebas de la casa para medir la intensidad de la señal

Lugar	Distancia	Señal	Calidad
Junto a la Raspberry	1 m	-44 dBm	Excelente
Sala	5 m	-59 dBm	Buena
Cocina	8 m	-66 dBm	Regular
Dormitorios	10 m	-69 dBm	Aceptable
Patio	13 m	-85 dBm	Mala

Resultado obtenido:

Las mediciones de intensidad de señal mostraron valores de -44 dBm junto a la Raspberry Pi, -59 dBm en la sala, -66 dBm en la cocina, -69 dBm en los dormitorios y -85 dBm en el patio. En todas las ubicaciones evaluadas el dispositivo Xiaomi 13T logró conectarse exitosamente a la red PiHoleWiFi, recibió correctamente una dirección IP.

Aunque la intensidad de la señal disminuyó conforme aumentó la distancia al punto de acceso, la conectividad permaneció estable y el sistema mantuvo el filtrado de contenido activo durante toda la prueba, demostrando una cobertura adecuada para un entorno doméstico.

4.1.10 PH-RO01: Rendimiento del SBC

Tabla 31

Rendimiento del SBC mediante pruebas de respuesta

Campo	Detalle

Objetivo	Evaluar el rendimiento de la Raspberry Pi 5 bajo condiciones de carga concurrente, verificando la capacidad de respuesta de la API REST y el consumo de recursos del sistema.
Precondición	Servicio api-parental.service activo en el puerto 8000, Pi-hole operativo, Suricata en ejecución, MariaDB disponible. Sistema en condiciones normales de operación.
Procedimiento	Se utilizó la herramienta Apache Benchmark (ab) para generar 100 solicitudes HTTP al endpoint /dispositivos/activos con una concurrencia de 10 clientes simulados. Durante la ejecución de la prueba se monitoreó el uso del procesador mediante htop y el consumo de memoria RAM con el comando free -m. Finalmente, se registraron las métricas de tiempo de respuesta, solicitudes procesadas por segundo y solicitudes fallidas obtenidas por Apache Benchmark.
Resultado esperado	El sistema tiene que procesar todas las solicitudes concurrentes con un tiempo de respuesta de menos de 500 ms, sin errores y una utilización de CPU y memoria que no comprometa la estabilidad de los servicios.

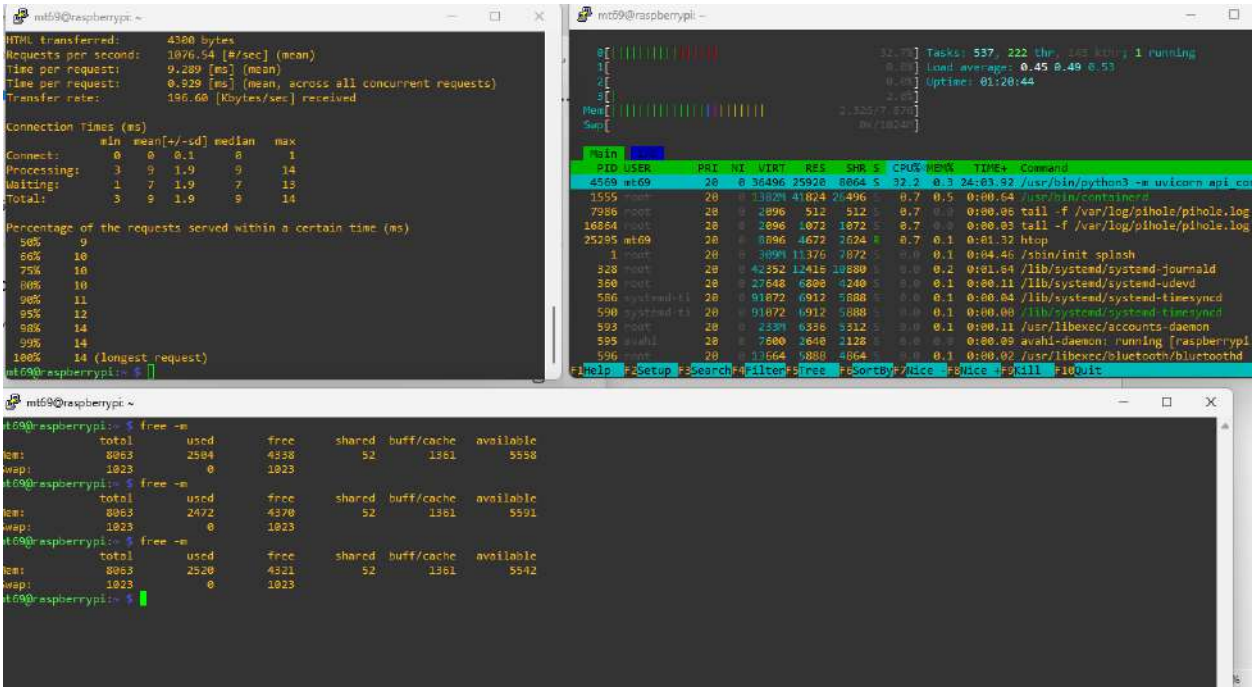
Se evaluó el rendimiento de la Raspberry Pi 5 bajo un escenario de carga concurrente utilizando la herramienta Apache Benchmark.

La prueba consistió en generar 100 solicitudes HTTP al endpoint /dispositivos/activos de la API REST, con una concurrencia de 10 clientes simulados, representando el acceso simultáneo de varios dispositivos de una red doméstica.

Durante la ejecución de la prueba se monitoreó el consumo de recursos del SBC mediante la herramienta htop, registrando el uso del procesador en tiempo real, mientras que el consumo de memoria RAM se verificó utilizando el comando free -m antes y después de la prueba.

Figura 97

Prueba de rendimiento del SBC dentro de un escenario controlado



Finalmente, se analizaron las métricas obtenidas por Apache Benchmark, incluyendo el tiempo promedio de respuesta, el número de solicitudes procesadas por segundo y la cantidad de solicitudes fallidas, con el propósito de determinar la capacidad de procesamiento del sistema.

Tabla 32

Resultados durante la prueba de rendimiento

Métrica	Resultado
Solicitudes concurrentes	10
Solicitudes totales	100
Tiempo promedio de respuesta	9.289 ms
Solicitudes por segundo	1076.54 req/s
Solicitudes fallidas	0
Uso máximo de CPU	32.2 %
Memoria utilizada	2520 MB (31 %)
Memoria disponible	5542 MB

Además, con el fin de someter al sistema a una condición de mayor exigencia, se realizó una prueba de estrés simulando 50 clientes concurrentes, con el propósito de evaluar su estabilidad y rendimiento bajo una carga elevada.

Figura 98

Prueba de rendimiento con 50 clientes simulados

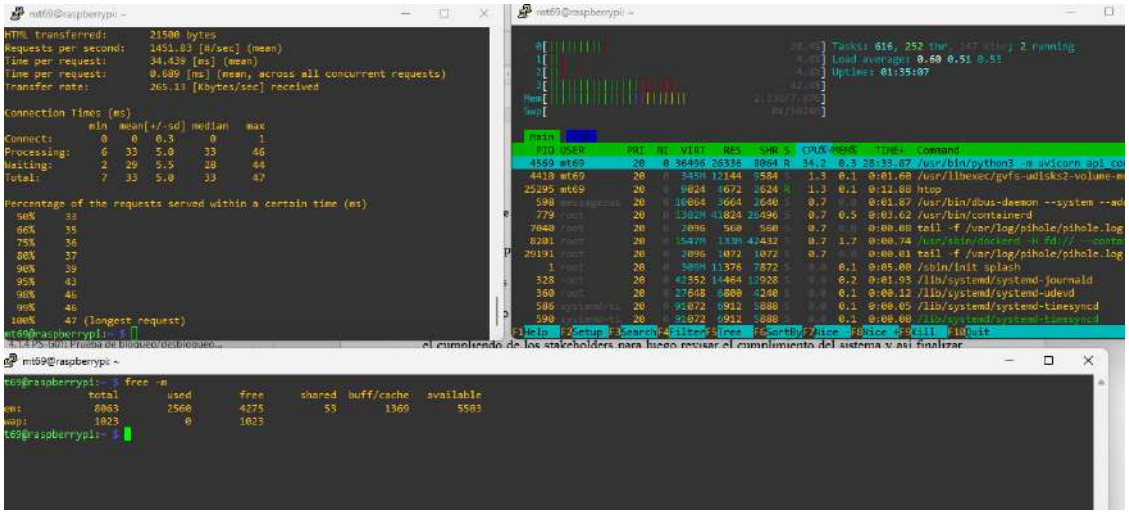


Tabla 33

Tabla comparativa con ambos escenarios de carga

Métrica	Carga media	Prueba de estrés
Solicitudes totales	100	500
Clientes concurrentes	10	50
Solicitudes procesadas por segundo	1076.54 req/s	1451.83 req/s
Tiempo promedio por solicitud	9.289 ms	34.439 ms
Solicitudes fallidas	0	0
Uso máximo de CPU	32.2 %	34.2 %
Memoria utilizada	2520 MB	2560 MB
Memoria total	8063 MB	8063 MB
Estado del sistema	Estable	Estable

Los resultados alcanzados exponen que el alza en el volumen de solicitudes concurrentes generó un incremento moderado en el tiempo de respuesta y en el uso de recursos del sistema. La Raspberry Pi 5, no obstante, mantuvo un comportamiento estable en ambos escenarios, sin mostrar solicitudes fallidas ni degradación del servicio. Esto demuestra que el prototipo es capaz de atender tanto una carga representativa de una red doméstica como una condición de mayor exigencia, conservando un rendimiento adecuado y un uso eficiente de los recursos disponibles.

Resultado obtenido

Durante la prueba de rendimiento se enviaron 100 solicitudes con una concurrencia de 10 clientes simulados mediante Apache Benchmark. La API procesó 1076.54 solicitudes por segundo,

con un tiempo promedio de respuesta de 9.289 ms y sin registrar solicitudes fallidas. El uso máximo del procesador fue de aproximadamente 32.2 %, mientras que el consumo de memoria alcanzó 2520 MB, equivalente al 31 % de la memoria total disponible. Estos resultados evidencian que la Raspberry Pi 5 mantiene un desempeño estable bajo carga concurrente y dispone de recursos suficientes para ejecutar simultáneamente FastAPI, Pi-hole, Suricata y MariaDB sin degradación perceptible del servicio.

4.1.10.1 Evaluación del rendimiento del hardware de la Raspberry Pi 5 bajo carga

Tabla 34

Prueba de rendimiento del SBC bajo carga

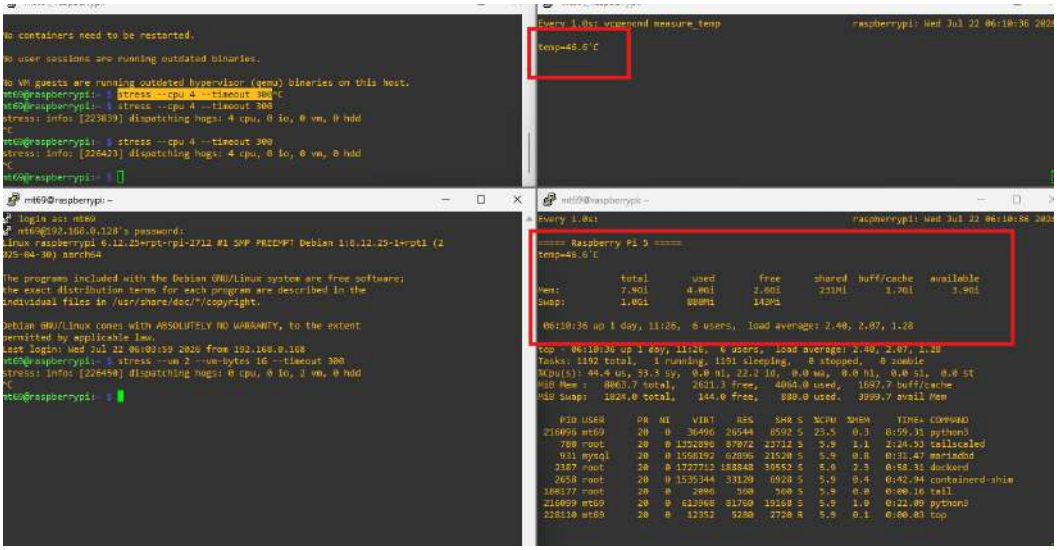
Campo	**Detalle**
Objetivo	Evaluar el comportamiento de la Raspberry Pi 5 al someter el procesador y la memoria RAM a condiciones de alta carga, verificando la estabilidad del sistema y el uso de recursos.
Precondición	Raspberry Pi 5 encendida con los servicios FastAPI, Pi-hole, Suricata y MariaDB en ejecución. Herramientas stress**,** top**,** free **y** vcgencmd instaladas y disponibles.
Procedimiento	Se registró el estado inicial del sistema mediante las herramientas top**,** free **y** vcgencmd para obtener el uso de CPU, memoria RAM y temperatura. Posteriormente, se ejecutó una prueba de estrés sobre los cuatro núcleos del procesador durante cinco minutos utilizando la herramienta stress. Finalizada esta prueba, se realizó una segunda prueba de estrés sobre la memoria RAM asignando dos procesos de 1 GB durante cinco minutos. Durante ambas pruebas se monitoreó continuamente la temperatura del procesador, el consumo de memoria

	y la carga del sistema para verificar su comportamiento bajo condiciones de alta demanda.
Resultado esperado	La Raspberry Pi 5 debe mantener un funcionamiento estable durante las pruebas de estrés, sin presentar bloqueos, reinicios inesperados ni temperaturas que comprometan la operación del sistema.

Se verificó el estado inicial del sistema, registrando la temperatura del procesador, el uso de memoria RAM y la carga del sistema.

Figura 99

Verificación del estado actual de la SBC antes de las pruebas



Se ejecutó una prueba de estrés sobre el procesador utilizando la herramienta stress, empleando los cuatro núcleos de la Raspberry Pi 5 durante cinco minutos. Durante la ejecución se monitoreó continuamente la temperatura del procesador mediante vcgencmd measure_temp, así como el comportamiento del sistema y el consumo de memoria utilizando las herramientas top y free. Posteriormente, se realizó una prueba de estrés sobre la memoria RAM, asignando dos

procesos que reservaron 1 GB de memoria cada uno durante un período de cinco minutos para evaluar la estabilidad del sistema bajo condiciones de alta carga.

Figura 100

Prueba de estrés a la SBC

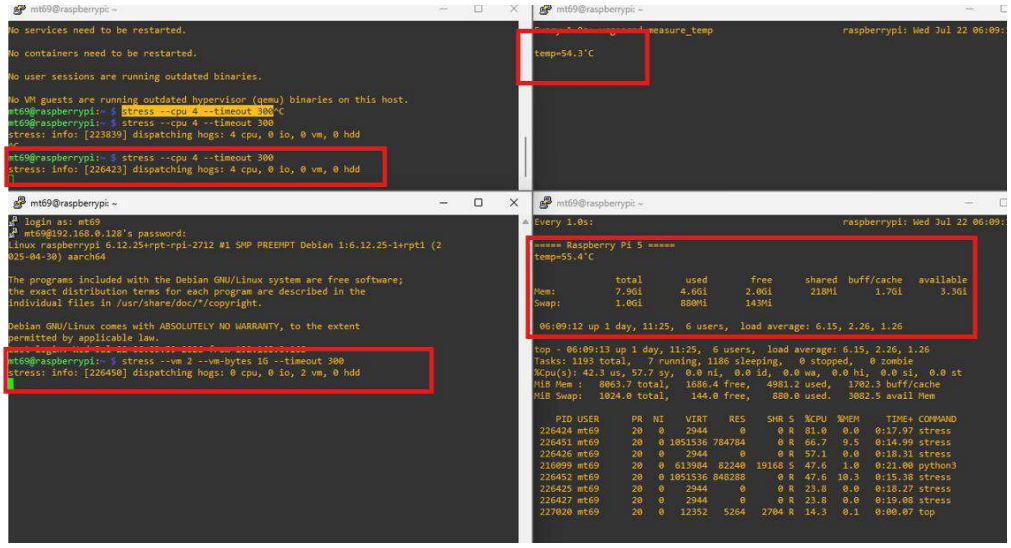


Tabla 35

Comparativa de eventos antes y durante la carga

Parámetro	Estado inicial	Estrés CPU	Estrés RAM
Temperatura	46.6 °C	54.3 °C	55.4 °C
Memoria utilizada	4.0 GB	4.6 GB	4.9 GB
Memoria disponible	3.9 GB	3.3 GB	3.0 GB
Load Average	2.40	6.15	6.15
CPU	Uso normal	Alta utilización (4 núcleos)	Uso moderado
Estado del sistema	Estable	Estable	Estable

Durante la prueba de estrés del procesador se observó un incremento de la temperatura desde 46.6 °C hasta aproximadamente 54.3 °C, manteniéndose dentro de los rangos normales de operación de la Raspberry Pi 5 y sin alcanzar temperaturas críticas. Se detectó un incremento en

el uso de la memoria RAM, dado que utilizando la herramienta stress se reservó alrededor de 2 GB, de forma que se redujo el total de memoria usable, pero la memoria me quedó disponible para el sistema operativo sin provocar fallas o interrupciones en los servicios correspondientes a los procesos ejecutados. El valor del Load Average también se vio alterado debido a la demanda artificial inducida en el CPU, lo que evidencia que se emplearon extensivamente los cuatro núcleos que tiene el procesador durante la prueba. De todas formas, el sistema continuó respondiendo correctamente y los servicios principales se mantuvieron funcionando.

Resultado obtenido

A lo largo de las pruebas realizadas sobre el procesador y sobre la memoria RAM, la Raspberry Pi 5 se mantuvo en estado estable, con un aumento controlado de la temperatura y el consumo de memoria. Los servicios del sistema permanecieron operativos en todo momento y no se presentaron bloqueos, reinicios ni fallos durante la ejecución de las pruebas, demostrando la capacidad del SBC para soportar cargas elevadas de procesamiento.

4.2 Cumplimiento de las pruebas

El cumplimiento de pruebas de realizaron con el propósito de comprobar que el prototipo cumple con las especificaciones establecidas durante las etapas de análisis y diseño del sistema. Por ello se hizo uso de los requerimientos específicos dentro del sistema. Primero se va a revisar el cumpliendo de los stakeholders para luego revisar el cumplimiento del sistema y así finalizar con el cumplimiento del diseño de la arquitectura.

4.2.1 Cumplimiento de requerimientos Stakerholders

Con el propósito de corroborar el funcionamiento correcto del prototipo desarrollado, se realizaron pruebas funcionales orientadas a validar el cumplimiento de los requerimientos

establecidos. Esto permite comprobar que cada una de las funciones establecidad responden a las necesidades planteadas en la Tabla 5.

Tabla 36

Cumplimiento de Requerimientos de Stakeholders

Requerimientos de Stakeholders	SI	NO	PARCIAL
StSR1	X		
StSR2	X		
StSR3	X		
StSR4	X		
StSR5	X		
StSR6	X		
StSR7	X		
StSR8	X		
StSR9	X		
StSR10			X
StSR11	X		
StSR12			X
StSR13	X		

El sistema actualmente cumple con todos los requerimientos de los stakeholders, pero la única cuestión aquí es que la StSR10 y StSR12 están de manera parcial, eso no quiere decir que no funcione al contrario esto se puede mejorar, en el StSR10 todavía existen limitaciones para definir reglas completamente personalizadas para cada situación o usuario. Por el momento trabaja

de forma general y no tiene sección de usuarios o clientes. El StSR12 si lo cumple, pero depende de otro sistema para que tenga salida a internet en este caso de una VPN.

4.2.2 Cumplimiento de requerimientos del sistema

La validación de los requerimientos del sistema se llevó a cabo mediante un conjunto de pruebas funcionales y de verificación. El objetivo fue comprobar que cada uno de los componentes propuestos cumplan con las especificaciones determinadas para el sistema. Estos requerimientos se encuentran en la Tabla 6.

Tabla 37

Cumplimiento de requerimientos del sistema

Requerimiento del sistema	SI	NO	PARCIAL
SySR1	X		
SySR2	X		
SySR3	X		
SySR4	X		
SySR5	X		
SySR6	X		
SySR7	X		
SySR8	X		
SySR9	X		
SySR10			X
SySR11	X		
SySR12	X		
SySR13			X
SySR14	X		

SySR15	X		
SySR16	X		
SySR17	X		
SySR18	X		
SySR19	X		

El sistema realizo diferentes pruebas de funcionamiento en el cual determino ciertas funciones que no pudieron ser completadas al 100%. En este caso se realiza unas observaciones a dichos requerimientos que necesitan optimización para que cumplan con lo requerido. El requerimiento SySR10 se puedo completar para 10 o más dispositivos en un rango de 10 metro.

Si bien el sistema logró asignar correctamente las direcciones IP y mantener la comunicación con los dispositivos conectados, se evidenció que el ancho de banda disponible y la capacidad del punto de acceso ocasionaron una disminución en el rendimiento de la red, generando inestabilidad y mayores tiempos de respuesta. Como resultado de las pruebas, se determinó que el funcionamiento óptimo del prototipo se alcanza con hasta cinco dispositivos conectados de manera simultánea, garantizando una conexión estable y un desempeño adecuado para las funcionalidades implementadas.

En cuanto al SySR13, tiene una limitación, al solo estar disponible para el SO Android. Esto hace que se otros sistemas operativos se queden sin este beneficio, lo que sería un problema. Pero para ello esta complementado la sección de bloqueo con DNS. Además, si se desinstala esta app está expuesta a evadir el sistema, ese sería una desventaja.

4.2.3 Cumplimiento de requerimiento de la arquitectura del sistema

La arquitectura fue diseñada con el fin de ser implementada de una manera continua, es decir que cada proceso dependa del anterior. Con ello podemos realizar pruebas en cada etapa e ir

descartando problemas. En esta parte se analiza el comportamiento tanto de manera de hardware y software que confirma el sistema. Esta referencia viene de la Tabla 7, donde se describe los requerimientos de la arquitectura.

Tabla 38

Cumplimiento de los requerimientos de la arquitectura

Requerimiento de la arquitectura del sistema	**SI**	**NO**	**PARCIAL**
SrSH1	X		
StSH2	X		
StSH3	X		
StSH4	X		
StSH5			X
StSH6	X		
StSH7	X		
StSH8	X		
StSH9	X		

Una vez determinado y puesto a prueba el sistema, se dio a conocer que el StSH5 tiene una pequeña limitación en cuanto a la cobertura y rendimiento de la red. Ya que como se mencionó anteriormente la red puede abarcar más 10 dispositivos, pero de manera limitada, con la velocidad reducida para cada dispositivo en la red.

Conclusiones

Se diseñó e implementó un prototipo de control parental basado en una plataforma SBC Raspberry Pi 5, integrando los servicios Pi-hole, Suricata IDS, FastAPI, MariaDB y una aplicación móvil, logrando centralizar la administración de dispositivos, el filtrado de contenidos y la supervisión de la actividad de red en un único sistema orientado a entornos domésticos.

La implementación del filtrado DNS mediante Pi-hole permitió controlar el acceso a dominios clasificados como inapropiados o potencialmente peligrosos. Durante las pruebas funcionales se verificó el bloqueo correcto de categorías como contenido adulto, apuestas y sitios maliciosos, demostrando que el mecanismo de listas negras funciona como primera capa de protección dentro de la red doméstica.

El desarrollo del algoritmo de análisis heurístico permitió complementar el filtrado tradicional mediante la evaluación del comportamiento de navegación. El sistema tuvo la capacidad de asignar grados de riesgo a los dispositivos conectados según diversos factores considerando, por ejemplo, intentos de acceso a determinados dominios restringidos, actividad nocturna, patrones de navegación, nivel de servicio, por tanto, puede generar alertas preventivas para el administrador.

La implementación de Suricata IDS posibilitó la detección de eventos relacionados con la actividad sospechosa dentro de la red, añadiendo esa información de tráfico de red a la que se recolectó a partir del análisis de la DNS. De este modo, se podían almacenar los eventos de seguridad y hacer uso de ellos posteriormente para la generación de reportes y el análisis del comportamiento de los dispositivos.

La aplicación para móviles desarrollada garantiza el manejo de las funciones básicas del sistema, tales como gestionar dispositivos, bloquear y desbloquear el acceso a Internet,

administración de los horarios del servicio, monitorización de las aplicaciones mediante ChildGuard, alertas y reportes automáticos. Esto facilitó la interacción del administrador con la plataforma sin requerir acceso directo a la Raspberry Pi.

La incorporación del modelo de inteligencia artificial permitió generar análisis interpretativos de los eventos recopilados por el sistema, transformando los registros técnicos de Pi-hole, Suricata y el motor heurístico en recomendaciones comprensibles para el usuario administrador, mejorando la capacidad de análisis frente a posibles riesgos.

Las pruebas de rendimiento validaron que la Raspberry Pi 5 puede considerarse como una plataforma para levantar el prototipo. En la prueba de carga con 10 clientes simulados concurrentes, la API alcanzó un throughput de 1076,54 peticiones por segundo, con un tiempo medio de respuesta de 9,289 ms y sin fallos en las peticiones. En la prueba de estrés, con 50 clientes simulados concurrentes, la API alcanzó un throughput de 1451,83 peticiones por segundo, con un tiempo medio de respuesta de 34,439 ms y sin comprometer la estabilidad del sistema.

Las pruebas de estrés del hardware corroboran que la Raspberry Pi 5 puede presumir la capacidad de procesar solicitudes bajo condiciones exigentes. Con la CPU y la memoria RAM a máxima carga, la temperatura, que partía de un límite de 46,6 °C, se aumentó a unos 55.4 °C, valores que quedaban dentro de rangos adecuados de operación, además de que los servicios principales continuaron funcionando sin interrupciones.

Las pruebas de cobertura inalámbrica permitieron validar que el punto de acceso que se había levantado a partir de la interfaz wlan0 proporcionaba conectividad suficiente en el entorno doméstico implementado. Las pruebas realizadas devolvieron unos valores que oscilan entre -44 dBm y -85 dBm, permitiendo que los dispositivos permanecieran conectados y utilizaran los servicios de filtrado DNS provistos por la Raspberry Pi 5.

Recomendaciones

De cara a futuras versiones del sistema, sería recomendable la utilización de modelos de inteligencia artificial ejecutados localmente, de manera que se puedan prevenir problemas derivados de la dependencia de servicios externos para el análisis, además de aumentar la privacidad de los datos analizados, mediante la implementación de algunos modelos de IA sobre una Raspberry Pi, o también sobre sistemas con mayor capacidad de procesamiento.

Se recomienda ampliar el análisis de seguridad mediante la incorporación de técnicas adicionales de inspección de tráfico cifrado y nuevos mecanismos de detección de amenazas, permitiendo mejorar la protección frente a ataques y comportamientos maliciosos emergentes.

Se sugiere incorporar nuevas funcionalidades en la aplicación móvil, como perfiles personalizados por usuario, estadísticas históricas de navegación, reglas avanzadas de control parental y configuraciones individuales según las necesidades del administrador.

Se debería realizar pruebas de escalabilidad con una cantidad superior de dispositivos conectados simultáneamente para evaluar el comportamiento del sistema en redes domésticas de mayor tamaño y determinar los límites operativos de la Raspberry Pi 5.

Es recomendable en todo momento mantener actualizadas las listas de filtrado utilizadas por Pi-hole, las reglas de detección de Suricata y los modelos de análisis inteligente, con el propósito de mejorar continuamente la precisión del sistema y su capacidad de identificar nuevas amenazas.

Bibliografía

[1] M. Guanotoa, “Implementación de un firewall de siguiente generación que minimice el riesgo que corren los niños al navegar por internet en una red doméstica”, 2024.

[2] INEC, “Tecnologías de la Información y Comunicación (TIC) en los hogares y en las personas 2020”, 2020.

[3] INEC, “Tecnologías de la Información y Comunicación (TIC) en los hogares y en las personas 2020”, 2020.

[4] A. Vallejo y M. Peñafiel, “Covid 19: Impacto en la educación ecuatoriana”, *Revista Científico Ecociencia*, vol. 9, pp. 76–92, dic. 2022, doi: 10.21855/ecociencia.90.755.

[5] B. B. A. López, “Sistema de monitoreo y control parental para redes de área local”, Tesis de grado, Universidad Técnica de Ambato, Ambato, Ecuador, 2021.

[6] R. S. Pressman, *Ingeniería del software Un enfoque práctico*, Séptima edición. McGRAW-HILL, 2010.

[7] François De Keersmaeker, Ramin Sadre, y Cristel Pelsser, *Supervising Smart Home Device Interactions: A Profile-Based Firewall Approach*. IEEE, 2024.

[8] ITU, “Directrices actualizadas para la protección infantil en línea: Recomendaciones para la industria”, 2020.

[9] UNESCO, “Reimaginar nuestros futuros juntos: un nuevo contrato social para la educación”, 2021.

[10] UNICEF, “Niños en un mundo digital: proteger la infancia en la era de internet”, 2020.

[11] UNICEF, “Niños en un mundo digital: proteger la infancia en la era de internet”, 2020.

[12] J. Assis y G. Valença, “Is My Child Safe Online?-On Requirements for Parental Control Tools in Apps used by Children”, *Journal on Interactive Systems*, vol. 15, núm. 1, pp. 823–838, ene. 2024, doi: 10.5753/jis.2024.4240.

[13] R. Tsai y L. Estell, “Network Activity for Parental Monitoring”, en *URTC 2024 - 2024 IEEE MIT Undergraduate Research Technology Conference, Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/URTC65039.2024.10937630.

[14] ONU, “Transformar nuestro mundo: la Agenda 2030 para el Desarrollo Sostenible”, 2015.

[15] William. Stallings, *Data and computer communications*, 10th ed. Pearson, 2014.

[16] B. A. Forouzan, *Data Communications and Networking*, 4th ed. New York: McGraw-Hill, 2007.

[17] J. F. . Kurose y K. W. . Ross, *Computer networking : a top-down approach*, 7th ed. Pearson, 2017.

[18] A. Vijayalakshmi, M. H. Lin, y R. N. Laczniak, “Evaluating Adolescents’ Responses to Internet Ads: Role of Ad Skepticism, Internet Literacy, and Parental Mediation”, *J. Advert.*, vol. 49, núm. 3, pp. 292–308, may 2020, doi: 10.1080/00913367.2020.1770639.

[19] S. Babu Erukala, D. Tokmakov, A. Devi Aguru, R. Kaluri, A. Bekyarova-Tokmakova, y N. Mileva, “An End-to-End Secure Communication Framework for Smart Homes Environment Using Consortium Blockchain System”, *IEEE Access*, vol. 13, pp. 67250–67268, 2025, doi: 10.1109/ACCESS.2025.3559070.

[20] I. K. Kovačevi´c, B. Štengl, y S. Groš, “Systematic review of automatic translation of high-level security policy into firewall rules”, Opatija: IEEE, may 2022, pp. 1063–1068. doi: 10.23919/MIPRO55190.2022.9803666.

[21] C. Gómez, L. Sepúlveda, y C. Candela, “Servidor Proxy Caché: Comprensión y asimilación tecnológica”, *INGE CUC*, vol. 8, 2012.

[22] N. Sangal, D. Singhvi, M. Pharande, y D. Patole, “Teen-alyse: A Mobile Application for Parental control, Teen Self-Monitoring and Active Mediation”, en *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), ICRITO 2021*, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/ICRITO51393.2021.9596148.

[23] E. Francisco, C. Moreno, G. Katherine, y C. Salazar, “Development of an Application for Parental Control of WhatsApp on Android Mobile Devices”, Quito, 2019. doi: 10.1109/ICI2ST.2019.00010.

[24] S. Ali, M. Elgharabawy, Q. Duchaussoy, M. Mannan, y A. Youssef, “Betrayed by the Guardian: Security and Privacy Risks of Parental Control Solutions”, en *ACM International Conference Proceeding Series*, Association for Computing Machinery, dic. 2020, pp. 69–83. doi: 10.1145/3427228.3427287.

[25] F. Fajardo, A. Supervisor, y M. Co-Supervisor, “Study on privacy of parental control mobile applications”, 2017.

[26] C. Pérez, C. Márquez, R. Silva, J. Sandoval, y J. García, “Una introducción a las Single Board Computers”, *Boletín Científico y Tecnológico UPIITA*, vol. 67, 2018.

[27] P. Sánchez, J. Jorquera, A. Huertas, G. Bovet, M. Gil Pérez, y G. Pérez, “A methodology to identify identical single-board computers based on hardware behavior fingerprinting”, *Journal of Network and Computer Applications*, vol. 212, mar. 2023, doi: 10.1016/j.jnca.2022.103579.

[28] S. J. Johnston *et al.*, “Commodity single board computer clusters and their applications”, *Future Generation Computer Systems*, vol. 89, pp. 201–212, dic. 2018, doi: 10.1016/j.future.2018.06.048.

[29] J. L. Álvarez, J. D. Mozo, y E. Durán, “Analysis of single board architectures integrating sensors technologies†”, el 1 de septiembre de 2021, *MDPI*. doi: 10.3390/s21186303.

[30] S. E. Mathe, H. K. Kondaveeti, S. Vappangi, S. D. Vanambathina, y N. K. Kumaravelu, “A comprehensive review on applications of Raspberry Pi”, el 1 de mayo de 2024, *Elsevier Ireland Ltd*. doi: 10.1016/j.cosrev.2024.100636.

[31] Fielding Roy, “Architectural Styles and the Design of Network-based Software Architecture”, 2000.

[32] V. Baladari, “HTTPS enforcement for distributed REST APIs for end-to-end data protection”, *International Journal of Business Quantitative Economics and Applied Management Research*, vol. 6, oct. 2019, doi: 10.5281/zenodo.15051683.

[33] D. ; S. J. ; K. M. Harrington, “ RFC 5592: Secure Shell Transport Model for the Simple Network Management Protocol (SNMP)”, 2009.

[34] F. Hauser, M. Haberle, M. Schmidt, y M. Menth, “P4-IPsec: Site-to-Site and Host-to-Site VPN with IPsec in P4-Based SDN”, *IEEE Access*, vol. 8, pp. 139567–139586, 2020, doi: 10.1109/ACCESS.2020.3012738.

[35] A. Abu Al-Hija’a, M. Andó, y B. J. Szekeres, “Secure command transmission techniques for industrial remote control”, *Sci. Rep.*, vol. 15, núm. 1, dic. 2025, doi: 10.1038/s41598-025-01290-x.

[36] J. R. C. D. C. Cruz, “OpenVProxy: Low Cost Squid Proxy Based Teleworking Environment with OpenVPN Encrypted Tunnels to Provide Confidentiality, Integrity and Availability”, Lima: IEEE, oct. 2020, p. 188.

[37] S. Komaragiri, A. Kuyya, N. Bandari, y J. Shinde, “Networkwide range ad-blocker using Raspberry Pi”, *International Research Journal of Modernization in Engineering Technology and Science*, jul. 2023, doi: 10.56726/irjmets43714.

[38] J. A. Donenfeld, “WireGuard: Next Generation Kernel Network Tunnel”, en *Proceedings 2017 Network and Distributed System Security Symposium*, Reston, VA: Internet Society, 2017. doi: 10.14722/ndss.2017.23160.

[39] Tailscale Inc., “What is Tailscale?”, What is Tailscale? [En línea]. Disponible en: https://tailscale.com/kb/1151/what-is-tailscale

[40] R. Perdigón Llanes, “Suricata como detector de intrusos para la seguridad en redes de datos empresariales”, *CIENCIA UNEMI*, vol. 15, núm. 39, pp. 44–53, may 2022, doi: 10.29076/issn.2528-7737vol15iss39.2022pp44-53p.

[41] A. Djenna, A. Bouridane, S. Rubab, y I. M. Marou, “Artificial Intelligence-Based Malware Detection, Analysis, and Mitigation”, *Symmetry (Basel).*, vol. 15, núm. 3, mar. 2023, doi: 10.3390/sym15030677.

[42] M. S. Toor, “Heuristic-Based Malware Detection For Android Using Machine Learning”, *International Journal of Trendy Research in Engineering and Technology*, vol. 08, núm. 06, pp. 14–19, 2024, doi: 10.54473/ijtret.2024.8603.

[43] W. Hazimah binti Wan Ismail, H. Ramadhani Mohd Husny, A. Syahman bin Mamat, y N. Ya Abdullah, “Parental Control System for Children on Wireless Network”, *Journal of Computing Technologies and Creative Content*, vol. 5, núm. 1, 2020.

Anexos

Anexo 1: Trabajos relacionados

Tabla 39 :

Ficha Bibliográfica de estudios con referencia a control parental y rede domesticas

Ficha de Referencia 1	
Estudio: Sistema de monitoreo y control parental para redes de área local	
Autor: Bryan Alejandro López Beltrán	**Editorial**: Universidad Técnica de Ambato
Año: 2021	**Ciudad**: Ambato
Ficha de Referencia 2	
Estudio: Implementación de un Firewall de Siguiente Generación que minimice el riesgo que corren los niños al navegar por internet en una red doméstica	
Autor: Guanotoa Chuma Alexander Mauricio	**Editorial**: Universidad Técnica del Norte
Año: 2024	**Ciudad**: Ibarra
Ficha de Referencia 3	
Estudio: Implementación de un servidor firewall-proxy bajo la plataforma de gnu/linux para la facultad de ingeniería en ciencias aplicadas, a fin de liberar procesamiento de los equipos del data center de la Universidad Técnica del Norte.	
Autor: Galo Israel Espinoza Padilla	**Editorial**: Universidad Técnica del Norte
Año: 2017	**Ciudad**: Ibarra

Ficha de Referencia 4

Estudio: Modelo de diseño de interfaz de usuario para aplicación de control parental en smartphones móviles utilizando el método de diseño centrado en el usuario	
Autor: Syafrizal Wardhana, Mira Kania Sabariah, Veronikha Effendy & Dana S. Kusumo	**Editorial**: Telkom University/ IEEE
Año: 2017	**Ciudad**: Bandung

Ficha de Referencia 5

Estudio: Actividad de red para la supervisión parental	
Autor: Romeo Tsai, Liam Estell	**Editorial:** McDaniel College / IEEE
Año: 2024	**Ciudad:** Montgomery Village

Anexo 2: Encuesta aplicada para el estudio del proyecto

Desde su perspectiva, ¿es pertinente desarrollar un sistema que contribuya a la protección de menores de 18 años frente a los riesgos asociados al uso de internet en el hogar?

- Si
- No

¿Qué dispositivos considera que los menores de edad utilizan con mayor frecuencia? (Puede marcar más de una opción)

- Teléfono celular
- Tableta
- Computadora
- Consola de video juegos

Televisión inteligente

En el hogar, ¿se supervisa actualmente el uso de internet por parte de los menores?

- Si
- No
- Aveces

¿Cuáles considera que son los principales riesgos del uso de internet en menores de edad? (Puede marcar más de una opción)

- Acceso a contenido inapropiado
- Publicidad engañosa
- Ciberacoso
- Uso excesivo de tiempo en línea
- Redes sociales sin control

En su hogar, ¿qué tipo de conexión emplean habitualmente los dispositivos para acceder a internet?

- Conexión inalámbrica (Wi-Fi)
- Conexión por cable (Ethernet)
- Red de datos móviles

¿Usted ha utilizado o utilizaría algún método para controlar el acceso a internet dentro del hogar?

- SI
- NO

¿Le interesaría un dispositivo y una aplicación que permita gestionar y limitar el uso de internet en el hogar?

- Si
- No
- Talvez

¿Qué funciones considera más importantes en un sistema de control parental? (puede marcar más de una opción)

- Bloqueo de páginas web
- Límite de tiempo de conexión a internet
- Reportes de actividad de navegación
- Bloquear publicidad engañosa
- Tener horarios de uso del internet

Considera que el sistema sistema debe ser fácil de instalar y manipular

- Totalmente de acuerdo
- De acuerdo
- Neutral
- En desacuerdo
- Totalmente en desacuerdo

¿Qué medio de administración preferiría para gestionar el sistema?

- Aplicación Móvil
- Página web
- Ambos

¿Cómo le gustaría que sea el dispositivo físico para el control de la red?

- Pequeño y discreto
- Similar a un router común

- Tamaño mediano y visible para control rápido
- No tengo preferencia

¿Qué tan importante es que la interfaz del sistema sea clara y fácil de entender, sin necesidad de conocimientos técnicos?

- Muy importante
- Importante
- Neutral
- Poco importante
- Nada importante

Resultados y análisis de la encuesta realiza en el sector establecido:

Pregunta 1: La encuesta refleja un contundente respaldo del 96,2% a favor del desarrollo de sistemas de control parental, evidenciando una alta preocupación y demanda por proteger a los menores de edad.

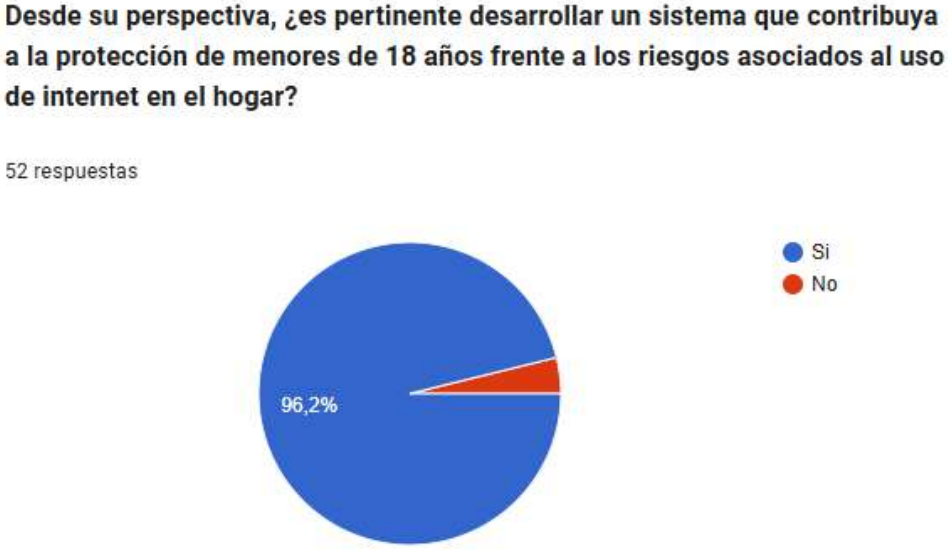


Figura 101 Resultados de la primera pregunta

Pregunta 2: El celular predomina como el dispositivo más utilizado por menores (84,6%), seguido por tabletas y computadoras, señalando que los sistemas de control parental deben priorizar entornos móviles.

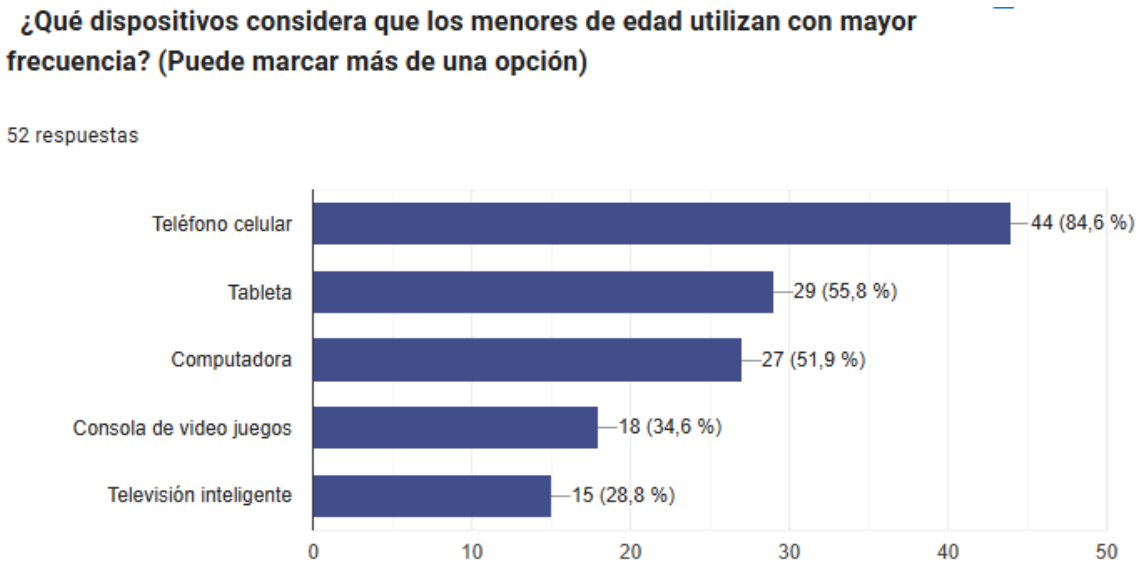


Figura 102 Resultados de la segunda pregunta

Pregunta 3: La mayoría de hogares no supervisa actualmente el uso de internet en menores (53,8%), revelando una brecha crítica de control que justifica implementar soluciones de supervisión accesible.

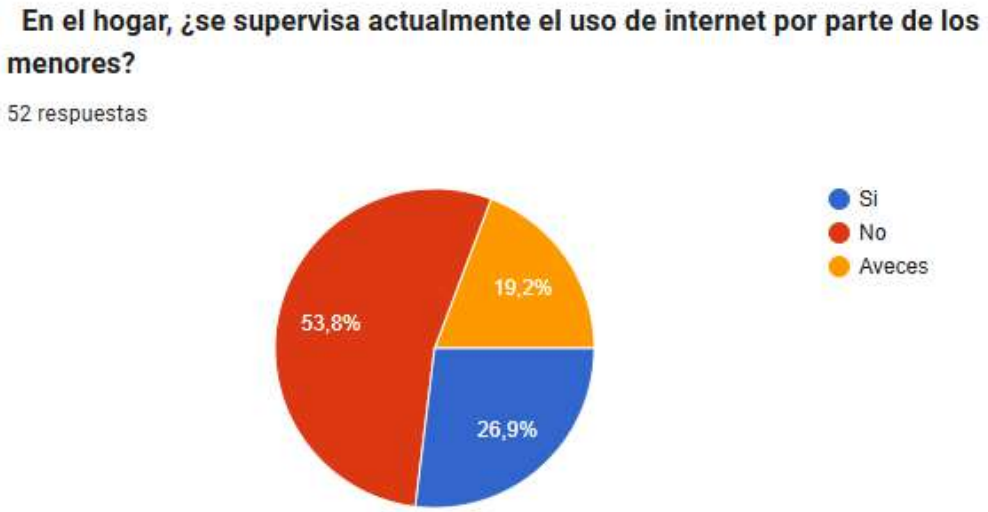


Figura 103 Resultados de la tercera pregunta

Pregunta 4: El acceso a contenido inapropiado lidera las preocupaciones con un 82,7%, seguido de cerca por la publicidad engañosa (73,1%), evidenciando que los riesgos percibidos abordan tanto la seguridad directa como la exposición comercial.



Figura 104 Resultados de la Cuarta pregunta

Pregunta 5: La conexión inalámbrica (Wi-Fi) domina ampliamente en los hogares con un 84%, superando al cable Ethernet (14%), lo que indica que el control parental debe ser efectivo sobre redes locales inalámbricas.

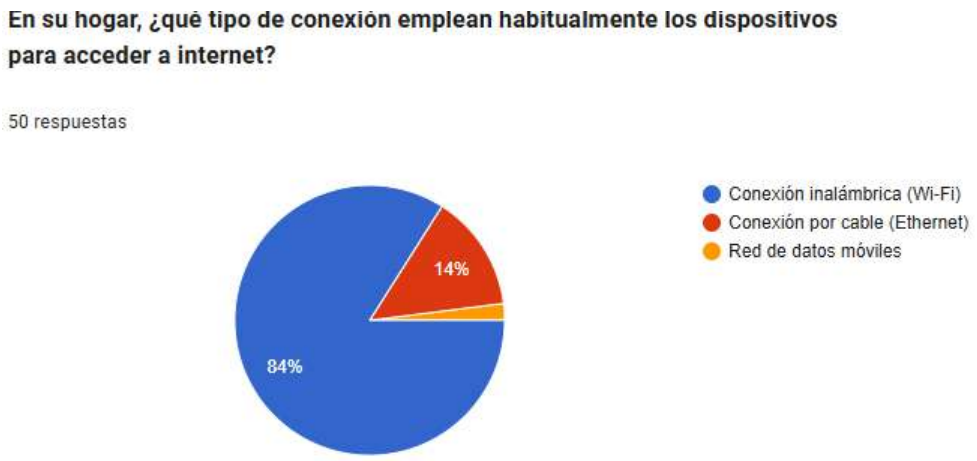


Figura 105 Resultados de la quinta pregunta

Pregunta 6: Existe una clara apertura a la adopción, pues el 53,8% afirma que ha utilizado o utilizaría algún método de control, mientras un notable 46,2% aún muestra reticencia a implementarlo en el hogar.



Figura 106 Resultados de la sexta pregunta

Pregunta 7: Un abrumador 90,4% de los encuestados demuestra un claro interés comercial en adquirir un dispositivo y aplicación para gestionar el internet, confirmando una alta viabilidad y demanda para esta propuesta de solución.

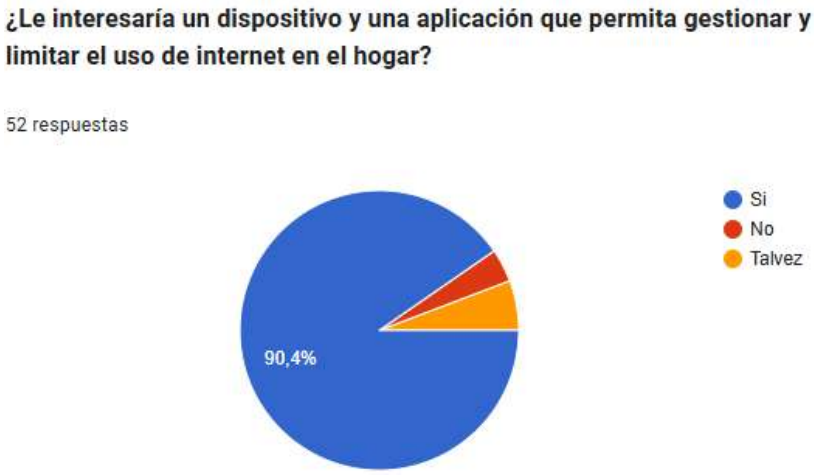


Figura 107 Resultados de la séptima pregunta

Pregunta 8: El límite de tiempo de conexión (69,2%) y el bloqueo de páginas web (67,3%) destacan como las funciones más valoradas, mostrando preferencia por herramientas directas de restricción de tiempo y contenido.

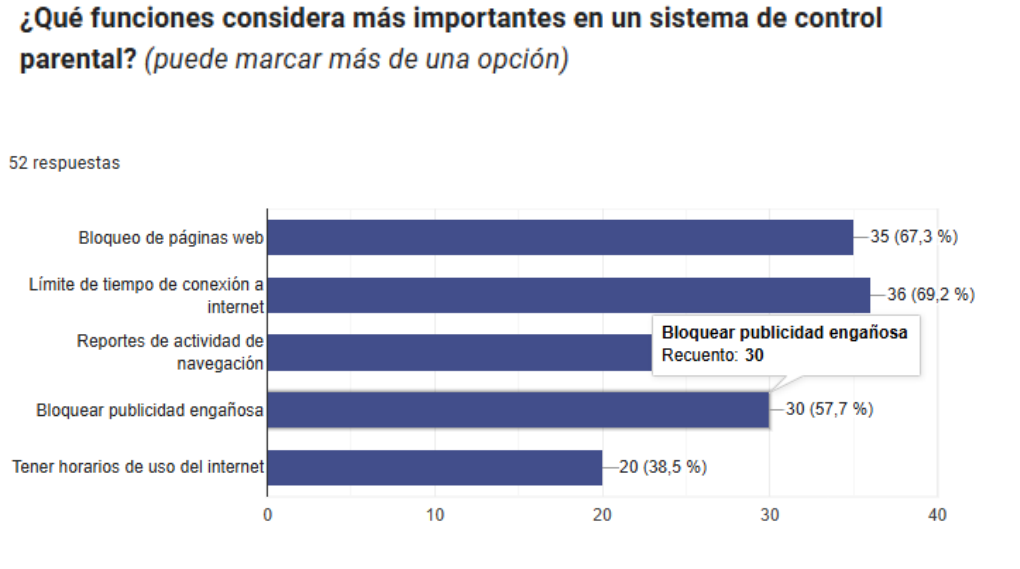


Figura 108 Resultados de la octava pregunta

Pregunta 9: La usabilidad es un factor decisivo, ya que el 82,7% acumulado está a favor 67,3% totalmente de acuerdo de que la solución sea sencilla de instalar y manejar.

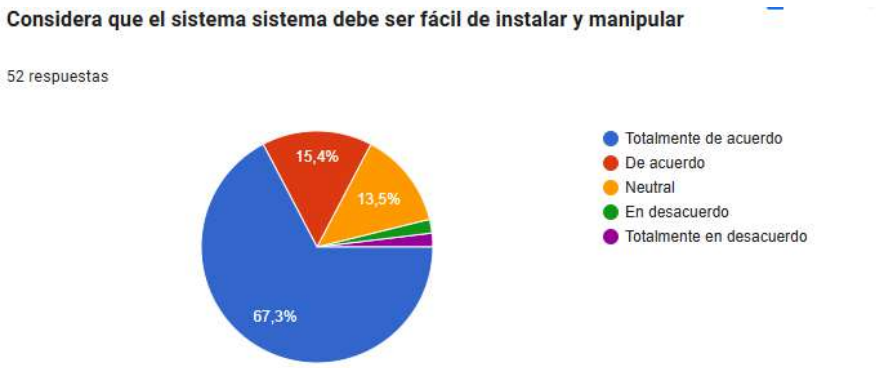


Figura 109 Resultados de la novena pregunta

Pregunta 10: Existe una marcada preferencia por la aplicación móvil como medio de administración principal (71,2%), reforzada por un 23,1% que optaría por combinarla con una página web.



Figura 110 Resultados de la décima pregunta

Pregunta 11: Los usuarios muestran una clara preferencia por el diseño del hardware, destacando un 65,4% que opta por un dispositivo "pequeño y discreto", superando significativamente a formatos tradicionales como el router.



Figura 111 Resultados de la 11va pregunta

Pregunta 12: La simplicidad en la interfaz es clave: un 82,7% considera "muy importante" que el sistema sea intuitivo y no requiera conocimientos técnicos, sumando un 96,2% total de aceptación.

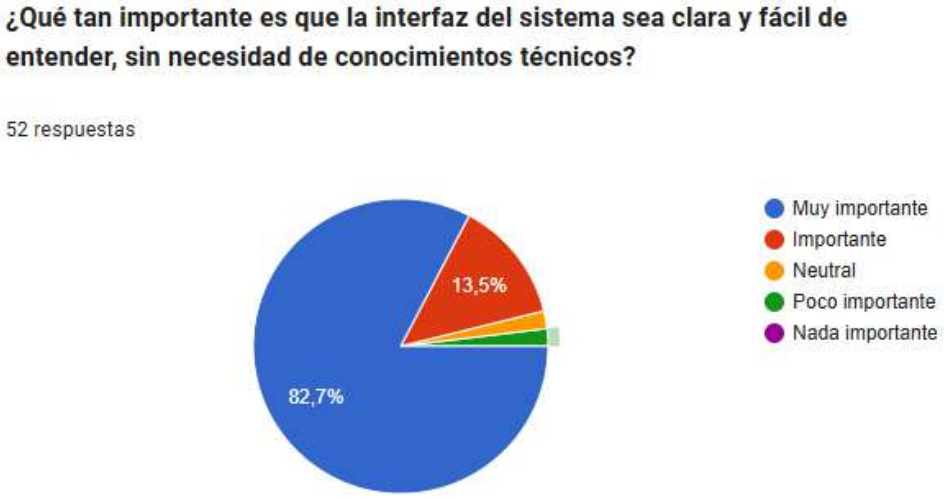


Figura 112 Resultados de la 12va pregunta

Anexo 3: Reporte general en PDF generado por el sistema

El reporte analiza el tráfico DNS, amenazas detectadas, nivel de riesgo por conductas de navegación y tiempo de uso de aplicaciones.

Sistema de Control Parental

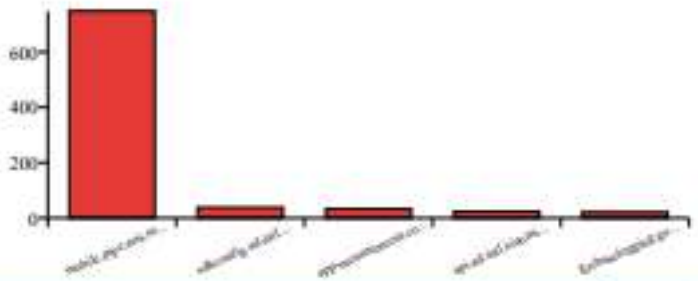
Reporte generado: 20/07/2026 19:18

Nivel de riesgo: Medio

Estadisticas del dia

Metrica	Valor
Total consultas DNS	3286
Dominios bloqueados	999
Porcentaje bloqueado	30.4%
Clientes activos	4
Dominios en lista negra	80874
Hora de mayor actividad	20:00

Top dominios bloqueados



Dominio	Consultas
mobile.pipe.aria.microsoft.com	748
sdkconfig.ad.intl.xiaomi.com	38
app-measurement.com	33
api.ad.intl.xiaomi.com	23
firebaselogging.googleapis.com	21
googleads.g.doubleclick.net	17
api16-access-wl-sg.pangle.io	17
crashlyticsreports-pa.googleapis.com	17

Analisis inteligente - Pachamama IA

Resumen: Hoy hubo una cantidad moderada de actividad en la red, con un total de 3286 consultas DNS y 999 dominios bloqueados. La mayor parte de la actividad se centró en plataformas de video como youtube.

Sitios frecuentes: youtube

Patron de uso: La hora de mayor actividad fue a las 20:00

Bloqueos del dia: Se bloquearon sitios como xvideos.com, firebaselogging.googleapis.com y googleads.g.doubleclick.net

Recomendaciones: Mantén una comunicación abierta con tus hijos sobre su actividad en línea, revisa regularmente los dispositivos conectados a la red y considera configurar sistemas de control parental para mayor seguridad.

Analisis Heuristico por Dispositivo

Evaluacion del comportamiento de navegacion de cada dispositivo en las ultimas 24 horas. Considera intentos a contenido prohibido, sitios peligrosos, apuestas, redes sociales y actividad nocturna (21:00-05:00).

Dispositivo: Xiaomi 13T

Metrica	Valor	Evaluacion
Nivel de riesgo	Alto	Score: 100/100
Intentos redes sociales	2	Normal
Intentos contenido adulto	8	Critico
Intentos apuestas	0	Normal
Intentos sitios peligrosos	8	Critico
Actividad nocturna (21-06h)	12	Fuera de horario

Dispositivo: MT

Metrica	Valor	Evaluacion
Nivel de riesgo	Medio	Score: 40/100
Intentos redes sociales	2	Normal
Intentos contenido adulto	3	Critico
Intentos apuestas	0	Normal
Intentos sitios peligrosos	3	Critico
Actividad nocturna (21-06h)	0	Normal

Dispositivo: Galaxy-A03

Metrica	Valor	Evaluacion
Nivel de riesgo	Bajo	Score: 0/100
Intentos redes sociales	0	Normal
Intentos contenido adulto	0	Bajo
Intentos apuestas	0	Normal
Intentos sitios peligrosos	0	Sin incidentes
Actividad nocturna (21-06h)	0	Normal

Dispositivo: Xiaomi-13T

Metrica	Valor	Evaluacion

Nivel de riesgo	Bajo	Score: 0/100
Intentos redes sociales	0	Normal
Intentos contenido adulto	0	Bajo
Intentos apuestas	0	Normal
Intentos sitios peligrosos	0	Sin incidentes
Actividad nocturna (21-06h)	0	Normal

Dispositivo: Saya

Metrica	Valor	Evaluacion
Nivel de riesgo	Bajo	Score: 0/100
Intentos redes sociales	0	Normal
Intentos contenido adulto	0	Bajo
Intentos apuestas	0	Normal
Intentos sitios peligrosos	0	Sin incidentes
Actividad nocturna (21-06h)	0	Normal

Estado de la Red y Amenazas Detectadas (Suricata IDS)

Analisis de red realizado el 20/07/2026 19:18. El sistema de deteccion de intrusiones Suricata monitorea el trafico de red en tiempo real con 51,894 reglas activas. Los eventos mostrados corresponden a las ultimas 24 horas de actividad en la red domestica.

Estado general de la red: **ATENCION REQUERIDA**

Tipo de evento: Escaneo de puertos

Que esta pasando en tu red: El evento indica un posible escaneo de puertos detectado por Suricata IDS, lo que puede ser un intento de identificar vulnerabilidades en la red

Acciones recomendadas:

- Revisar los registros de Suricata para obtener mas informacion
- Bloquear el trafico proveniente de la direccion IP asociada al escaneo
- Verificar la configuracion de firewall y seguridades de red

Detalle tecnico de eventos:

Evento	Protocolo	Nivel
1:9000001:1] ESCANEO DE PUERTOS DETECTADO - Posible Nmap	TCP	Alta

Nota: El analisis heuristico de comportamiento (PI-Hole) y el monitoreo de red (Suricata IDS) trabajan en conjunto para ofrecer una vision completa de la actividad en la red domestica.

ChildGuard - Control de Apps del Hijo

Informacion recopilada por la aplicacion ChildGuard instalada en los dispositivos de los hijos. Muestra el uso de aplicaciones, apps bloqueadas y actividad reciente.

Dispositivo: Saya

Metrica	Valor
Estado de conexion	En linea

Bateria actual	24%
Ultima app usada	launcher
Apps instaladas	43
Apps bloqueadas	0

Apps mas usadas hoy:

Aplicacion	Tiempo de uso
Avatar World	1h 59min
YouTube	49 minutos
Facebook	20 minutos
Inicio de One UI	3 minutos
WhatsApp	3 minutos

Apps instaladas recientemente (ultimos 3 dias):

Aplicacion	Fecha de instalacion
WPS Office	2026-07-20 19:10
Buscador	2026-07-20 19:10
TikTok Lite	2026-07-20 19:10
Service provider location	2026-07-20 19:10
Google	2026-07-20 19:10

Recomendacion: No hay apps bloqueadas. Revisa las apps instaladas y considera bloquear las que no sean apropiadas para la edad de tu hijo.

Registro de Ataques e IPs Bloqueadas

Ataques detectados hoy por Suricata IDS:

Fecha	IP Origen	Tipo de Ataque
2026-07-20 02:20	192.168.0.168	Ataque SYN Flood detectado
2026-07-20 02:20	192.168.0.168	Escaneo de puertos detectado
2026-07-20 02:20	192.168.0.168	Ataque SYN Flood detectado
2026-07-20 02:20	192.168.0.168	Escaneo de puertos detectado

No hay IPs bloqueadas actualmente. La red no registro ataques bloqueados manualmente.
